Dpto. de Teoría de la Señal y Comunicaciones
Universidad Carlos iii de Madrid

TESIS DOCTORAL

Blancos Blandos Enfatizados para Clasificación Máquina

Autor: Soufiane El Jelali
Directores: Dr. Aníbal Ramón Figueiras Vidal
Dr. Abdelouahid Lyhyaoui

Leganés, 2011

Tesis Doctoral:
Blancos Blandos Enfatizados para Clasificación Máquina

Autor:
Soufiane El Jelali

Directores:
Dr. Aníbal Ramón Figueiras Vidal
Dr. Abdelouahid Lyhyaoui

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores

Presidente:

Vocales:

Secretario:

acuerda otorgarle la calificación de

Leganés, a

Agradecimientos

Quiero agradecer a todas las personas que han estado a mi lado, y que han sido participes para llevar a cabo esta Tesis Doctoral durante mi estancia de estudios del doctorado en el Departamento Teoría de la Señal y Comunicaciones de la Universidad Carlos III de Madrid, en especial,

A mi tutor Dr. Aníbal Ramón Figueiras-Vidal por sus ideas, su eficacia, su compromiso y su disponibilidad para dirigir generosamente esta Tesis Doctoral, así, dándome la oportunidad de aprender y descubrir el mundo de las máquinas de aprendizaje.

A mi co-director Dr. Lyhyaoui Abdelouahid por su participación en los trabajos de investigación de esta Tesis.

A todo el profesorado del Programa del Master Interuniversitario Multimedia y Comunicaciones de la Universidad Carlos III de Madrid por su calidad de formación que me ha permitido consolidar mis conocimientos para llevar a cabo los trabajos de investigación.

A todos los miembros del departamento Teoría de la Señal y Comunicaciones que han puesto su grano de arena directamente o indirectamente en esta Tesis.

A mis compañeros de los laboratorios 4.2.C.01 y 4.2.C.03 (Rafa, Manu, Vanessa, Miguel, Jaisiel, Carlos, Efraín, Adil, Luis, Roberto, Sergio, Ricardo, Neila, Marilea, Marta, y Anas), sin olvidar, los compañeros que han compartido conmigo las clases del Master -mencionado arriba- y en especial Rocío por haberme ayudado mucho en el primer año.

A todos mis amigos por sus consejos y sus afectos, a todos los profesores que me han enseñado en mi vida académica, y a todos que, un día, han pensado en mí.

A la Agencia Española de Cooperación Internacional por haber financiado mis estudios del Master y mi primer año del doctorado, y a la UC3M por la beca P.I.F ("Personal Investigador en Formación") que me ha permitido finalizar los trabajos correspondientes a dicha Tesis.

Finalmente, a mis padres, a mi hermano, y a mis familiares por el apoyo constante que he recibido de ellos.

A mis padres

Resumen

El modo más habitual de entrenar máquinas de clasificación es minimizar mediante búsqueda analítica una función de coste que depende de los valores de blancos y de salidas. Ello impone abandonar las salidas duras, no derivables. Además, el carácter discreto de los blancos no permite obtener buenos diseños considerando salidas lineales.

De lo anterior surge la conveniencia de emplear las clásicas activaciones sigmoidales; ahora bien, su presencia no puede considerarse "natural" para cualesquiera problemas.

Por otra parte, la ponderación de los errores entre blanco y salida de la máquina es una técnica bien conocida que permite conceder más atención a aquellos ejemplos que resulten más importantes para un buen aprendizaje. Esa importancia típicamente es función creciente del correspondiente error y de la proximidad a la frontera de decisión de las muestras, aunque en forma dependiente del problema y no conocida "a priori".

Lo dicho conduce a concebir la posibilidad de construir y aplicar blancos blandos enfatizados ("Emphasized Soft Targets", ESTs): valores modificados de los blancos, en principio distintos para distintos ejemplos, establecidos según la relevancia de cada ejemplo para el aprendizaje. Con ello, cabe la posibilidad de prescindir de la activación, aprovechando el carácter "continuo" de los ESTs, al tiempo que el énfasis facilita obtener diseños de buenas prestaciones. Debe resaltarse que la supresión de la activación permite utilizar para clasificación formulaciones que son propias de la estimación, como es el caso del modelado directo de las muestras mediante mezcla de gaussianas ("Gaussian Mixture Models", GMM) y, sobre todo, de los llamados procesos gaussianos ("Gaussian Processes", GP), versión generalizada del filtro de Wiener y que presenta numerosas ventajas de manejo e interpretación frente a métodos alternativos de regresión no lineal.

La presente Tesis explora la utilización de ESTs para resolver problemas de clasificación, considerando tanto esquemas tradicionales -perceptrones multicapa ("Multi-Layer Perceptrons", MLPs) entre los discriminativos, GMMs entre los generativos- como los ya mencionados GPs. Se presenta y aplica una poderosa forma de ESTs, consistente en una combinación convexa local del blanco original y la salida de un clasificador auxiliar o guía, siendo el parámetro funcional de combinación dependiente del error y la proximidad a la frontera de cada muestra tratada por la guía. Los resultados obtenidos indican que estos ESTs permiten frecuentemente alcanzar mejores (y muy altas) prestaciones, si bien a cambio de un sensible inconveniente de carga computacional debido a la necesidad de determinar mediante validación cruzada ("Cross Validation", CV) los valores de los parámetros de la forma de los ESTs. Versiones simplificadas llevan a situaciones intermedias.

Una sostenida reflexión sobre el desarrollo del trabajo y los resultados de éste condujo a determinar una semejanza funcional inmediata entre ponderaciones de errores y ESTs, así como a una interpretación del papel de las activaciones desde la perspectiva de regulación de la atención que se dedica a las diferentes muestras. Ello abre la posibilidad de recurrir a conversiones de ponderaciones a ESTs y de ESTs a ponderaciones en una serie de situaciones en que cabe esperar ventajas -mejora de prestaciones o simplificación de arquitecturas-, así como de disponer de orientación para elegir formas de las activaciones. Tales posibilidades se examinan y discuten en el Capítulo 6, y las más atractivas se incluyen como sugerencia de líneas futuras, junto con otras y la revisión de las aportaciones de la Tesis, en el último capítulo.

Abstract

The most frequent training of classification machines consists on minimizing, by means of an analytical search, a cost function which depends on targets and output values. It does not allow the use of hard outputs, that are not derivable. Furthermore, the discrete character of the classification targets does not permit to get good designs with linear outputs. The importance of introducing the classical sigmoidal activations emerges from the above facts; but it is clear that these activations cannot be considered as "natural" for all the classification problems.

On the other hand, weighting output errors is a well known technique that serves to pay more attention to those examples that are more relevant for a good learning. This relevance is usually related with the corresponding error and with the proximity to the decision border of each sample, although in a previously unknown and problem dependent manner.

The previous facts suggest the possibility of constructing and applying Emphasized Soft Targets (ESTs): Modified values for the targets, basically different for different examples, and defined according to the relevance of each labeled sample for the learning process. In this way, it is possible to avoid the presence of the nonlinear activation, because the ESTs are "continuous", and, simultaneously, the effect of the emphasis helps to obtain a good performance. We remark that suppressing the nonlinear activation permits to develop classifier designs that are based on regression models, such as the direct form of Gaussian Mixture Models (GMMs), and, mainly, the so-called Gaussian Processes (GPs), a generalized version of the famous Wiener filter, which offers many working and interpretation advantages in comparison with alternative nonlinear regression methods.

This Thesis explores the use of ESTs to solve classification problems, considering both traditional machines -Multi-Layer Perceptrons (MLPs) among the discriminative family, GMMs among the generative schemes- and GPs. A powerful form of ESTs is introduced and applied; it consists on a local convex

combination of the original target and the output of an auxiliary classifier, or "guide". The functional combination parameter depends on each sample's error and its proximity to the border according to the auxiliary classifier. A lot of experimental results support the hypothesis of that ESTs frequently allow to get a better (and very high) performance, although paying a significant increase of the training computational effort, due to the need of carrying out Cross Validation (CV) processes to establish the values of ESTs parameters. Simplified versions of the proposed EST forms offer intermediate levels of compromise.

Thinking all the time on the work being developed and its results led to find an immediate functional similarity between error weighting and ESTs, as well as to an interpretation of the role of nonlinear activations from the perspective of controlling the degree of attention to different examples. This opens the way to converting sample weighting methods to ESTs and viceversa in a series of situations that promise advantages when doing so (better performance or even simpler architectures). Also, the said perspective on the role of the nonlinear activations gives a guide to select their forms. All these possibilities are considered and discussed in Chapter 6, and the most promising of them are included as suggestions of new research lines, along with other opportunities and a resume of contributions, in the final chapter.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este capítulo revisamos algunos conceptos fundamentales del aprendizaje máquina, haciendo especial hincapié sobre decisión y clasificación. Citamos las principales dificultades para la obtención de la debida generalización en el diseño de las máquinas de aprendizaje, destacando los métodos de Selección y Edición de Muestras como posibilidades para resolverlas. Cerramos el capítulo presentando los objetivos y un resumen del contenido de esta Tesis.

1.1. Los problemas de clasificación

En un problema de clasificación, el objetivo es dividir el espacio de entrada en regiones, cada una asociada a una determinada clase. La tarea de clasificación se realiza mediante funciones discriminantes $g_j(\mathbf{x})$ que asignan la muestra $\mathbf{x}$ a la j^*-ésima clase C_{j^*} si

$$j^* = \arg \max_j g_j(\mathbf{x}) \tag{1.1}$$

Existen dos perspectivas para abordar un problema de clasificación: la analítica y la aproximación máquina.

La analítica, de base estadística, conduce a las teorías bayesiana y frecuentista de decisión [Van Trees1968]. Se utiliza cuando se dispone de información

estadística (verosimilitudes $p(\mathbf{x}|H_i)$, parámetros de coste $C_{ji} = C(D_j, H_i)$ con $0 < C_{ii} < C_{ji} \;\; \forall\, j \neq i$; para el caso bayesiano, también las probabilidades *a priori* $P(H_i)$) (véase Fig 1.1).

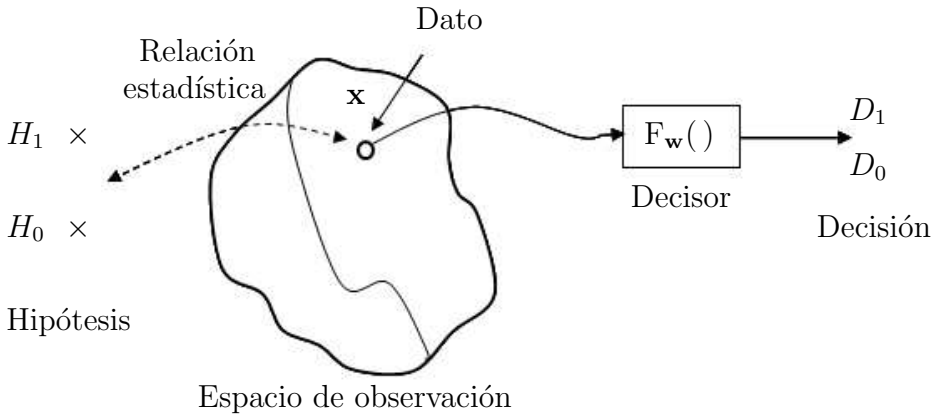

Figura 1.1: Visión analítica de la decisión para un problema binario.

Bajo la formulación bayesiana, el objetivo es diseñar un clasificador de óptimas prestaciones, es decir, obtener una decisión D_{j^*} que minimiza el coste medio a la vista de $\mathbf{x}$,

$$j^* = \arg \min_j \overline{C}(D_j|\mathbf{x}) \tag{1.2}$$

con

$$\overline{C}(D_j|\mathbf{x}) = \sum_{i=1}^{I} C_{ji} P(C_i|\mathbf{x}) \tag{1.3}$$

siendo $P(C_i|\mathbf{x})$ la probabilidad *a posteriori* de la clase C_i condicionada a $\mathbf{x}$, que se calcula a partir del teorema de Bayes

$$P(C_i|\mathbf{x}) = \frac{P(C_i)\, p(\mathbf{x}|C_i)}{\sum_{i=1}^{I} P(C_i)\, p(\mathbf{x}|C_i)} \tag{1.4}$$

donde I es el número de clases.

En muchos casos se pueden elegir los parámetros de coste C_{ji} $(i, j = 1, \ldots, I)$ como sigue: $C_{ii} = 0$ (no hay coste asociado a una clasificación correcta) y $C_{ji} = c$, $j \neq i$; c (> 0) es una constante; todos los errores de clasificación tienen el mismo coste c asociado; en este caso, $\overline{C}(D_j|\mathbf{x})$ se simplifica a

$$\overline{C}(D_j|\mathbf{x}) = c \sum_{j \neq i} P(C_i|\mathbf{x}) = c(1 - P(C_j|\mathbf{x})). \tag{1.5}$$

En este caso, el clasificador óptimo que minimiza $\overline{C}(D_j|\mathbf{x})$ lleva a maximizar la probabilidad *a posteriori* $P(C_j|\mathbf{x})$:

$$j^* = \arg \max_j P(C_j|\mathbf{x}) \tag{1.6}$$

que se denomina clasificador MAP ("Maximum A Posteriori") con $g_j(\mathbf{x}) = P(C_j|\mathbf{x})$ como función discriminante. Dicho clasificador proporciona la mínima probabilidad de error.

En la práctica, no es habitual disponer de esta información estadística. Es posible recurrir a diseños semianalíticos que estiman esa información a partir de los datos disponibles. En lo que se refiere a la estimación de densidades de probabilidad (ddps), hay tres tipos de aproximaciones:

- Paramétricas: se asume una forma analítica de la función de ddp y se estiman sus parámetros partiendo de los datos disponibles. De los métodos de estimación paramétrica destacan el de máxima verosimilitud (ML, "Maximum Likelihood")(una discusión formal sobre el origen de la idea ML se encuentra en [Akaike1973]; véase también [Bishop2006]) y los métodos basados en la inferencia Bayesiana (estos métodos se pueden revisar en [MacKay1991, MacKay2003]). La estimación ML considera los parámetros como valores fijos pero desconocidos, y la mejor estimación de cada parámetro corresponde al valor que maximiza la verosimilitud de todos los datos de entrenamiento. Por el contrario, la estimación bayesiana ve cada parámetro de la ddp como una variable aleatoria que tiene una forma conocida de distribución *a priori*, y se calcula su densidad *a posteriori* dado el

conjunto de datos con la fórmula de Bayes. Naturalmente, el conocimiento de la forma de las ddps no es usual, y adoptar una forma errónea se paga en diseños de bajas prestaciones.

- No paramétricas: se aplica un modelo general capaz de aproximar la distribución de acuerdo con las observaciones sin necesidad de hacer ninguna asunción sobre la forma de la ddp de los datos. Los k vecinos más próximos (k-NN, "k-Nearest Neighbours") y las Ventanas de Parzen [Duda2001] son las técnicas más utilizadas. Desafortunadamente, estas técnicas demandan muchos recursos computacionales en memoria y cálculo.

- Semiparamétricas: son modelos flexibles, típicamente combinaciones convexas de ddps, que, tras ajustar los parámetros, permiten reducir el coste computacional de las técnicas no paramétricas. Destacan los Modelos de Mezcla de Gaussianas (GMMs, "Gaussian Mixture Models"), cuyos parámetros se optimizan empleando el algoritmo EM, "Expectation-Maximization" [Dempster1977].

Amplias discusiones de todos estos métodos se encuentran en [Fukunaga1990, Duda2001].

Una segunda posibilidad de proceder a partir de observaciones es la aproximación máquina, consistente en parametrizar una función discreta $F_{\mathbf{w}}(\mathbf{x})$ con datos etiquetados, dividiendo según los valores de $F_{\mathbf{w}}$ el espacio de entrada en regiones que se asocian a una determinada clase. Entre otras máquinas, cabe señalar: las Redes Neuronales, NNs, "Neural Networks" (MLP, "Multi-Layer Perceptron", RBFNN, "Radial Basis Function Neural Network") [Haykin1999, Duda2001, Bishop2006]; los métodos basados en núcleos [Schölkopf2002](Procesos Gaussianos, GPs, "Gaussian Processes" [Rasmussen2006], Máquinas de Vectores Soporte, SVMs, "Support Vector Machines" [Vapnik1995, Burges1998]); los Sistemas Expertos [Michalsky1980] y los Árboles de Decisión [Breiman1984]; y los modelos gráficos, como, por ejemplo, las Redes Bayesianas [Buntine1994].

1.2. Diseño y generalización

Los algoritmos convencionales de entrenamiento de una máquina de decisión minimizan funciones de coste para aproximar la probabilidad de error teórica, ya que no hay una correspondencia directa entre las fórmulas teóricas para clasificación y posibles implementaciones máquina que se pueden entrenar mediante búsqueda local. Aunque hay algoritmos que aproximan la tasa de error empírica, como la Regla del Perceptrón [Rosenblatt1958], ofrecen una mala generalización y presentan dificultades de convergencia. Fisher propuso formulaciones [Fisher1936] que usan medidas de separación para diseñar decisores lineales. Los Algoritmos Basados en Decisión ("Decision Based Algorithms" [Kung1995]) son una aproximación con ciertas semejanzas a la Regla del Perceptrón, así como el uso de Funciones de Energía ("Energy Functions" [Telfer1994]). Persiguiendo el mismo objetivo de minimizar la probabilidad de error, se han propuesto los métodos de Máximo Margen (MM, "Maximum Margin"), que conducen al diseño de las SVMs [Boser1992, Cortes1995, Schölkopf1995, Müller2001].

Estas aproximaciones están estrechamente relacionadas con la capacidad de generalización de las máquinas. Se dice que una máquina generaliza bien cuando acierta en el cálculo de la relación entrada-salida para muestras no vistas en la fase de entrenamiento. Para conseguirlo, resulta legítimo que las máquinas de aprendizaje incorporen, durante el proceso de diseño, mecanismos que favorezcan la capacidad de generalización.

En ese sentido, es interesante la idea de modificar los algoritmos convencionales de entrenamiento para que la máquina preste más atención a las muestras que son importantes para reducir la probabilidad de error de clasificación y, así, proporcione una definición apropiada de la frontera de decisión; esto es el objetivo de las técnicas de Gestión de Muestras (GM). Los métodos de GM ofrecen varias ventajas: permiten, entre otras cosas, acelerar la convergencia, reducir el coste computacional, y asegurar una buena generalización. La filosofía en que se fundamentan estos métodos es diferenciar la contribución de los datos durante el proceso de entrenamiento según convenga para una buena clasificación, mediante mecanismos que focalicen el entrenamiento en las muestras más importantes. En

el segundo capítulo revisamos algunos de los principales trabajos desarrollados en este ámbito.

Las técnicas de GM sirven para compensar el carácter subóptimo de los métodos convencionales de entrenar clasificadores máquina. El primer trabajo en esta línea de investigación apareció a finales de los años 60 [Hart1968], y fue seguido por varios esquemas y técnicas propuestas para enfatizar el entrenamiento en las muestras de acuerdo a su cercanía a la frontera de decisión [Sklansky1980, Plutowski1993, Choi2002] o en función de la medida del error [Munro1992, Cachin1994]. Otros trabajos propusieron nuevos criterios de GM, pero con el objetivo de determinar los centros de máquinas RBF [Lyhyaoui1999]. La formulación original de los algoritmos de Boosting [Freund1996a, Freund1996b, Schapire1999] minimiza una función de coste exponencial basada en el margen, y en cada ronda las muestras erróneas se enfatizan de acuerdo con una función de error de ese tipo. En [Gómez-Verdejo2006, Gómez-Verdejo2008] se demuestra que la función de énfasis del Real AdaBoost se puede descomponer en el producto de dos términos, uno relacionado con el error cuadrático de las muestras y otro asociado a su proximidad a la frontera; esto permite introducir un parámetro para regular la combinación de los dos criterios. En general, no está claro cuál de los dos tipos de muestras es más importante para obtener un buen diseño, aunque la respuesta parece depender del problema que se considere [Franco2000].

Una alternativa interesante consiste en reemplazar la clasificación basándose en las etiquetas originales por otra que use una versión blanda, porque eso permite introducir énfasis sobre las muestras por medio del diseño de los blancos blandos, al tiempo que puede ponerse en transformar el problema en uno de estimación (regresión), para el que los costes típicos resultan naturales. Hay varias maneras de generar etiquetas blandas, por ejemplo el suavizado convolucional [Reed1992, Reed1995] o la reducción selectiva de las etiquetas [Mora-Jiménez2009]; pero parece sensato potenciar las ventajas ofrecidas por los métodos de énfasis. Esta es la orientación de [Gorse1997] y de nuestros trabajos: en [El Jelali2008a] se ha presentado una nueva propuesta basada en una combinación convexa entre las etiquetas originales y la salida de una guía auxiliar tipo MLP. En dicho artículo se propone una forma de énfasis aplicada sobre los blancos que:

- Presenta una estructura suficientemente general (y, consiguientemente, de alto potencial), en el sentido de que permite enfatizar parcialmente y teniendo en cuenta la distancia a la frontera y el tamaño del error indicados por un clasificador auxiliar;

- Incluye tres parámetros ajustables, lo que le dota de la posibilidad de adaptar su forma concreta al problema que se está considerando; si bien hace necesario recurrir a la Validación Cruzada (CV, "Cross Validation"), que supone un notable incremento de la carga computacional necesaria para llevar a cabo el aprendizaje;

- Mantiene los valores del blanco blando resultante entre -1 y 1, lo que permite observar si se produce degradación al pasar de la formulación de decisión (clasificación) a la de estimación (regresión), comparando los resultados obtenidos sin insertar e insertando una activación convencional.

Los resultados obtenidos en [El Jelali2008a] acreditan que, manejando la forma de énfasis propuesta -que se presentará en detalle más adelante-, no hay diferencias relevantes entre utilizar una activación o no: lo que sirve para validar la posibilidad de utilizar formulaciones para estimación cuando se aplique dicho énfasis (u otros análogos) a los blancos. En [El Jelali2008b, El Jelali2009] se amplían dichos resultados, al tiempo que se aplica el procedimiento a otras formulaciones, como los GMMs y, de manera preliminar, a los GPs.

1.3. Objetivos y organización de la Tesis

1.3.1. Objetivos

En la presente Tesis se propone la construcción de blancos blandos enfatizados (EST, "Emphasized Soft Targets") para reemplazar la etiqueta (dura) deseada y, adicionalmente, formular y resolver los problemas de decisión (clasificación) como problemas asociados de estimación (regresión).

Se examinan los casos convencionales de diseños MLP y GMM, por su valor intrínseco y como análisis preliminares a la aplicación de dicho procedimiento a

la clasificación mediante GPs; lo que constituye el objetivo principal de la Tesis.

La razón para fijar ese objetivo radica en las ventajosas características de los GPs empleados para estimación (regresión):

- La posibilidad de recurrir a versiones que incluyen pocos parámetros entrenables y, además, la de entrenar éstos mediante Máxima Verosimilitud (ML, "Maximum Likelihood"), lleva a diseños altamente resistentes al sobreajuste (sin tener que recurrir a técnicas complementarias);

- Los estimadores GP proporcionan también de forma inmediata una medida de la fiabilidad de los resultados obtenidos.

Cuando se desea diseñar un clasificador GP, se hace necesario, si no se desea caer en el alto coste computacional que implica emplear procedimientos Monte-Carlo, introducir una variable latente, a través de la cual y el uso de una activación sigmoidal se puede derivar la probabilidad "a posteriori" de una de las hipótesis y, por tanto, resolver el problema de decisión (clasificación). Sin embargo, los métodos correspondientes han de recurrir a algoritmos que permitan parametrizar el modelo GP subyacente mediante la aproximación de integrales intratables; ello supone un no despreciable aumento de la carga computacional necesaria para el diseño. Aunque cabe la posibilidad de incorporar procedimientos de énfasis en estos procesos, la complejidad resultante puede resultar prohibitiva -salvo para esquemas de énfasis muy simples-, ya que añadirá la necesidad de CV para buscar los parámetros del énfasis sobre una ya elevada carga computacional. Por ello, y a la vista de la ya mencionada posibilidad de convertir los problemas de decisión (clasificación) en otros de estimación (regresión) mediante la creación de blancos blandos con un énfasis convenientemente flexible y potente, obteniendo buenas prestaciones, la exploración de su empleo para GPs tiene suficiente importancia como para dedicarle un análisis detallado.

Además de lo anterior, durante el desarrollo de los trabajos de la Tesis se evidenció la existencia de sencillas relaciones directas entre las técnicas de GM y los procedimientos EST. Esas relaciones, que abren nuevas perspectivas de ambos tipos de procesos, se exponen y discuten en el capítulo que precede a las conclusiones de la Tesis.

1.3.2. Organización de la Tesis

El resto de los capítulos de esta Tesis se organiza según se dice a continuación.

El segundo capítulo se divide en dos partes: en la primera se revisan algunos trabajos relevantes de GM. La segunda parte se dedica a presentar la propuesta de énfasis llamada "Blancos Blandos Enfatizados" (ESTs, "Emphasized Soft Targets") o "Etiquetas Blandas Enfatizadas".

En el tercer capítulo, tras un breve recordatorio de los MLPs, se presenta el diseño de clasificadores MLPs con ESTs mediante la elección de una razonable forma para el parámetro de la combinación convexa; corroborando la calidad de los diseños mediante una serie de experimentos.

La extensión a los GMMs del procedimiento anterior se hace en el Capítulo 4 utilizando la formulación discriminativa para estimar las etiquetas blandas bajo modelado conjunto de éstas y los datos. Una serie de experimentos avalan los buenos resultados de este método.

En el quinto capítulo se extiende el método de énfasis a los GPs, como alternativa a las aproximaciones que son necesarias para llevar a cabo la tarea de clasificación con ellos y su elevado coste computacional. El capítulo incluye una parte experimental que demuestra el potencial del uso de los ESTs.

En el sexto capítulo se formaliza la relación de los ESTs con los ya tradicionales procedimientos de ponderación de términos de errores muestrales, discutiendo desde esta nueva perspectiva los resultados obtenidos y exponiendo las posibilidades que queden abiertas.

Finalmente, en el capítulo de Conclusiones se exponen éstas y se indican líneas futuras de I+D para extender estos trabajos.

Capítulo 2

Blancos Blandos

2.1. Introducción

Los métodos de GM se basan en modificar los algoritmos convencionales prestando más atención a las muestras relevantes para el proceso de aprendizaje, a fin de que aporten la información más influyente en la determinación de la frontera de decisión. En el algoritmo estándar el entrenamiento se efectúa dando la misma importancia a todas las muestras de entrenamiento; pero la contribución de las observaciones en el aprendizaje no debe ser igual: algunos ejemplos pueden resultar redundantes o bien no ayudan a definir las fronteras de clasificación; al contrario, otros pueden ser decisivos para obtener una buena solución. Entre las ventajas que ofrecen las técnicas de GM podemos citar la reducción del coste computacional del entrenamiento, además de facilitar una buena generalización.

En lo que sigue, no se pretende revisar todos los trabajos realizados sobre la GM, sino centrarse en aquéllos que han inspirado aspectos de esta Tesis, o bien que puedan servir para ampliar lo que aquí se propone.

2.2. Gestión de Muestras

El trabajo de Hart [Hart1968] fue el punto de partida para desarrollar otros métodos de GM con distintos criterios y diferentes objetivos. Hart considera que las muestras erróneas y cercanas a la frontera de decisión son importantes para el aprendizaje. Su algoritmo, CNN ("Condensed Nearest Neighbor"), es como sigue:

1. Dividir el conjunto de entrenamiento original en dos subconjuntos, "Store" y "Grabbag". En principio, se mete la primera muestra en el primer subconjunto, y el resto de las muestras en el segundo.

2. Clasificar cada muestra en "Grabbag" mediante un algoritmo 1-NN usando el subconjunto "Store" como conjunto de entrenamiento. Si la muestra resulta mal clasificada, se mete en "Store".

3. Repetir la etapa 2 hasta que "Grabbag" quede vacío o no haya muestras a meter en "Store".

4. Devolver el subconjunto "Store" como resultado de la selección.

[Cachin1994] adoptó diversas estrategias para la selección de patrones basadas en la medida del error. Las máquinas DBNN ("Decision Based Neural Networks") de Kung y Taur [Kung1995] consideran solamente muestras erróneas durante el entrenamiento. En [Zhang1994a] se presentó un procedimiento para seleccionar las muestras altamente erróneas durante el entrenamiento. Munro [Munro1992] propuso repetir el entrenamiento de las muestras erróneas hasta la convergencia. [Strand1992] es otro ejemplo de esta clase de procedimientos que focalizan el entrenamiento en un subconjunto formado por muestras difíciles de aprender. Incluso la formulación básica de los esquemas de Boosting parece basarse en el error de todas las muestras [Freund1996a, Freund1996b, Schapire1999].

Por otra parte, Sklansky et al. [Sklansky1980] postulan que la proximidad a la frontera es un aspecto esencial para seleccionar las muestras relevantes, proponiendo un método para construir una frontera lineal a tramos, empleando

discriminantes lineales, mediante la identificación de los pares opuestos más cercanos. En [Lyhyaoui1999] se presentan más esquemas que seleccionan muestras próximas a la frontera aplicados al diseño de RBFNNs. Otros autores siguieron la misma línea en sus trabajos [Cheung1992, Plutowski1993, Choi2002].

Las formulaciones originales de los esquemas de Boosting para construir conjuntos de máquinas [Freund1996a, Freund1996b, Schapire1999] se basan en dar importancia a las muestras erróneas, aunque se ha demostrado que el Real AdaBoost enfatiza las muestras erróneas y también aquéllas que están cerca de la frontera [Gómez-Verdejo2006, Gómez-Verdejo2008]; la última referencia presenta algunas generalizaciones. El algoritmo de Máximo Margen usado para entrenar las SVMs [Boser1992, Cortes1995, Müller2001] puede considerarse como un procedimiento que presta atención a los dos tipos de muestras. No está claro qué tipo de muestras es importante para el aprendizaje, salvo para cada problema [Franco2000].

A continuación, revisamos varios métodos importantes de GM aplicados a los MLPs.

2.2.1. GM para entrenamiento de MLPs

A finales de los años 80, en varios trabajos [Denker1987, Huyser1988] se demostró, para un MLP entrenado con el algoritmo BP ("Back Propagation") bajo error cuadrático medio (MSE, "Mean Square Error") como función de coste, que emplear un conjunto formado por muestras localizadas cerca de la frontera de decisión puede suponer una buena generalización.

Técnicas basadas en muestras confusas

Wann [Wann1990] demostró que no se garantiza buena generalización usando un conjunto de entrenamiento de gran tamaño, sino mediante un subconjunto formado por muestras cercanas a la frontera de clasificación, seleccionado de modo adecuado. Para ello, Wann utiliza el criterio del vecino más próximo para distinguir entre dos tipos de muestras:

- *Típicas*: lejos de la frontera de decisión, que no dan lugar a confusión.
- *Muestras cerca de la frontera y que generan confusión*: si una muestra perteneciente a una clase tiene al menos un vecino próximo de la otra clase de entre un cierto número -bajo- de vecinos más próximos, es probable que dicha muestra esté localizada al borde de la frontera.

Para determinar estos dos tipos de muestras, la cercanía de una muestra del conjunto original de datos a la frontera de decisión se mide de acuerdo al número de vecinos más próximos de la clase contraria. De acuerdo con la idea de Wann, el conjunto original de datos se divide en cuatro subconjuntos A, B, C, y D, según el número de vecinos más próximos de la clase contraria: A no tiene ninguno (subconjunto de muestras típicas), B uno, C dos, y finalmente, D tres, de entre los tres vecinos más proximos.

Wann construyó el conjunto de entrenamiento con 14 combinaciones de estos subconjuntos para estudiar la influencia de éstos sobre la capacidad de generalización del MLP, y concluyó que la máquina puede generalizar bien si se entrena solamente con las muestras cercanas a la frontera de clasificación, y la presencia de los subconjuntos C y D en el conjunto de entrenamiento es suficiente para que la red generalice bien. Además, las redes entrenadas con subconjuntos de muestras cercanas a la frontera de decisión excluyendo los subconjuntos C o D o los dos a la vez no ofrecen una mejora con respecto a redes entrenadas con muestras típicas.

Continuando con la misma idea de muestras típicas y confusas, Ohnishi [Ohnishi1991] introdujo cuatro métodos de presentación de las muestras para aumentar la eficacia del aprendizaje de un MLP. Examinó cuatro estrategias diferentes para acelerar la convergencia:

1. Presentación de las muestras típicas y luego de las muestras confusas.
2. Presentación alternada de muestras típicas y confusas de una en una.
3. Presentación solamente de las muestras típicas.
4. Presentación solamente de las muestras confusas.

Los resultados demuestran que las tres primeras estrategias dan mejores resultados; por otro lado, la cuarta da peores resultados con respecto al método convencional (entrenar con todo el conjunto de entrenamiento en orden aleatorio). Parece una contradicción con los dos trabajos citados anteriormente, a pesar de seguir la misma filosofía: pero el diseño del MLP y los problemas de clasificación tratados en los trabajos son diferentes, y eso lleva a concluir que esta técnica es sensible a las características del problema.

Muestras difíciles de aprender

En el año 1992, algunos trabajos [Cheung1992, Munro1992, Strand1992] estudiaron el entrenamiento de un MLP empleando muestras difíciles de aprender.

En [Cheung1992] se divide el aprendizaje en tres fases en función del comportamiento del error cuadrático medio a lo largo de las iteraciones del entrenamiento. Las tres etapas del proceso de aprendizaje son las siguientes:

1. **Fase de convergencia del error:** el error cuadrático de las muestras del conjunto de entrenamiento tiende a un valor estable; esto ocurre al principio del proceso de aprendizaje.

2. **Fase de competencia:** inmediatamente después, el entrenamiento global obedece a las muestras que cumplen la condición $\nabla^{\mathrm{T}}\mathrm{E}\nabla\mathrm{E}_k > 0$; $\nabla\mathrm{E} = \sum_k \nabla\mathrm{E}_k$ y $\nabla\mathrm{E}_k$ son el gradiente del error cuadrático de todas las muestras y de la k-ésima muestra, respectivamente.

3. **Fase de dominación:** en esta etapa, algunas muestras dominan el entrenamiento, es decir, sus errores decrecen más rápidamente ("dominantes") con respecto a las otras ("no dominantes"): las muestras "dominantes" participan más que las "no dominantes" en la determinación de la frontera de clasificación; la información que aportan las muestras "no dominantes" se capta al principio del aprendizaje. Esa dominación desaparece tras un cierto número determinado de iteraciones; si no, el error cuadrático de todas las muestras queda atrapado en un mínimo local.

De la discusión anterior se concluye que el tiempo de aprendizaje y la variación del error dependen del comportamiento de las muestras durante el aprendizaje. No todas se comportan de la misma manera: algunas son fáciles de aprender, mientras que otras ponen dificultades al aprendizaje; esto puede conducir a una degradación en la capacidad de generalización y desacelerar la convergencia del proceso de aprendizaje. Cheung propuso dos variantes del algoritmo BP convencional fijando la atención en las muestras que presentan dificultades de aprendizaje durante la etapa de dominación; así se consigue reducir el error y mejorar las prestaciones. Dichas variantes son:

- **Conjunto Dinámico de Entrenamiento (CDE)**

  Se trata de una solución que pretende aumentar la frecuencia de presentación de las muestras con dificultades de ser aprendidas. Se aumenta el conjunto de entrenamiento dinámicamente con ejemplos que dan lugar a un error cuadrático mayor que el error cuadrático medio del conjunto de entrenamiento, con el fin de aportar una nueva información al aprendizaje; este nuevo conjunto de entrenamiento se denomina CDE.

  Sea P el conjunto de entrenamiento original. El procedimiento CDE queda descrito del siguiente modo:

  1. Entrenar la red con P y determinar T, el conjunto de muestras con mayor error.

  2. Construir el conjunto CDE incluyendo P y réplicas de todas las muestras en T hasta alcanzar un tamaño prefijado.

  3. Entrenar la red con el conjunto CDE.

  4. Actualizar el CDE: Considerando los errores de la última clasificación, buscar en P la muestra i que tiene el máximo error. Buscar en T la muestra j que tiene el mínimo error. Si $j = i$ añadimos una réplica de la muestra i a CDE; si no, reemplazamos una réplica de la muestra j por la i.

  5. Repetir los pasos 3 y 4 hasta alcanzar un criterio de parada.

- **Factor de Ponderación (FP)**

  Este método modifica la dirección del algoritmo de búsqueda BP en cada iteración en función del error cuadrático de las muestras durante el entrenamiento, con el fin de evitar que el error cuadrático se quede atrapado en un mínimo local.

  Sea $\mathbf{g}$ la dirección de búsqueda de un BP bloque convencional, $\mathbf{g} = -\sum_k \nabla \mathrm{E}_k$, siendo E_k el error cuadrático de la k-ésima muestra $\mathbf{x}^{(k)}$. El método FP modifica el termino $\mathbf{g}$ introduciendo un factor de ponderación f_k proporcional a E_k; la nueva expresión de la dirección de búsqueda es: $\mathbf{g}_{\text{nueva}} = -\sum_k f_k \nabla \mathrm{E}_k$. El funcionamiento del método se describe como sigue: si una muestra $\mathbf{x}^{(k)}$ presenta dificultades de aprendizaje, entonces $\mathbf{g}_{\text{nueva}} \nabla \mathrm{E}_k < 0$; por supuesto, E_k es grande, por tanto f_k tomará un valor grande hasta que el producto $\mathbf{g}_{\text{nueva}} \nabla \mathrm{E}_k$ se vuelva positivo y E_k disminuya.

Parece que las dos variantes de Cheung introducen un coste computacional adicional debido al crecimiento del conjunto de entrenamiento en el método CDE y al aumento de las operaciones a efectuar sobre la dirección de búsqueda del BP convencional en el método FP. No obstante, logran aumentar la velocidad del aprendizaje con respecto al algoritmo convencional y que el coste computacional neto no sea mayor.

Strand [Strand1992] siguió la misma idea de enfocar el entrenamiento en las muestras con dificultades de aprendizaje, pero su criterio es diferente: en este caso, el método es decremental y consiste en formar un conjunto de entrenamiento activo con las muestras que no han sido aprendidas por la red; las otras muestras aprendidas al principio del proceso de aprendizaje se excluyen después de un cierto tiempo del conjunto activo para evitar la redundancia de la información. Se eliminan del conjunto activo aquellas muestras que satisfacen la condición $\mathrm{E}_k < \text{máx}\,(J, K)\,\mathrm{E}/K^2$, siendo E_k, E, J y K el error cuadrático de la muestra $\mathbf{x}^{(k)}$, el error total del conjunto original de entrenamiento, la J-ésima época y el número de muestras de entrenamiento, respectivamente. Por otra parte, las muestras descartadas en la época J se meten en una cola y vuelven para refrescar el entrenamiento en la época $J+1$. A veces no resulta tan fácil descartar algunas muestras que pueden ser imprescindibles para el aprendizaje, lo que llevaría a

pensar en otro criterio para detectar las muestras críticas. También conviene señalar que es conveniente conservar las ddps condicionales de las clases a la hora de eliminar muestras y así tener estabilidad estadística.

Munro [Munro1992] sigue la misma estrategia de aprendizaje que Cheung y Strand, aumentando la frecuencia de presentación de las muestras en cada época hasta que el error es menor que un cierto umbral β. La incógnita de esta propuesta es que no se puede saber si el tiempo de convergencia aumentará o disminuirá, dependiendo del problema de clasificación. Además, el algoritmo es muy sensible al valor del umbral β y a la complejidad del problema a tratar.

Método de selección incremental

Zhang publicó varios trabajos de GM [Zhang1991a, Zhang1991b, Zhang1993a, Zhang1993b, Zhang1994a, Zhang1994b]. Las dos últimas contribuciones engloban los trabajos anteriores, en los cuales define un método incremental de GM que parte de un subconjunto D_{K_0} (de K_0 muestras) del conjunto de entrenamiento original $D_K = \{(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})\}_{k=1}^{K}$ ($K_0 < K$), y a lo largo del entrenamiento se añaden poco a poco nuevas muestras que tienen error cuadrático relativamente grande.

Al principio, el método divide el conjunto de entrenamiento original D_K en dos subconjuntos: D_{K_0}, un subconjunto de entrenamiento formado por un número pequeño K_0 de muestras escogidas de D_K de manera aleatoria, y C, un subconjunto de muestras candidatas, formado por las muestras restantes. El procedimiento incremental es el siguiente:

1. Entrenar la red con el conjunto D.
2. Calcular la salida de la red correspondiente a las muestras de C y seleccionar un cierto número de muestras $\{(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})\}$ del subconjunto C que tienen los mayores errores cuadráticos.
3. Actualizar los subconjuntos C y D:
   $D_{\text{nuevo}} \Leftarrow D_{\text{antiguo}} \cup \{(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})\}$, $C_{\text{nuevo}} \Leftarrow C_{\text{antiguo}} - \{(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})\}$.

4. Repetir los pasos de 1 a 3 hasta alcanzar un criterio de parada o que el subconjunto C se quede vacío.

El propio Zhang [Zhang1994b] modifica el algoritmo anterior, aplicando procesos incrementales sobre el conjunto de entrenamiento y el tamaño de la red simultáneamente. El algoritmo determina de manera constructiva el tamaño óptimo de la red; así, por un lado, evita los problemas de convergencia por escasez de unidades en las capas ocultas, y por otro lado, no cae en el problema de sobre-entrenamiento por exceso de las mismas. Plutowski et al. [Plutowski1993] propusieron un método similar orientado a resolver problemas de regresión.

Trabajos relacionados con la anterior técnica son [Fukumizu1994, Ishibuchi1994, Marchand1993, Yamasaki1994].

Estrategias pedagógicas de Cachin

Cachin [Cachin1994] propuso varias estrategias para la GM que denominó Estrategias Pedagógicas de Selección de Patrones ("Pedagogical Pattern Selection Strategies"). Las ideas básicas de dichas estrategias son, por una parte, centrar el entrenamiento en las observaciones que generan un gran error, y por otra, refrescar el entrenamiento de vez en cuando con las muestras que son fáciles de aprender. Cachin definió las siguientes estrategias:

- **Probabilidad de presentación dependiendo del error ("Error-Dependent Presentation Probability" -EDPP):**

  La presentación de las muestras es aleatoria; sin embargo, la distribución de probabilidad de presentación al entrenamiento de las muestras no es uniforme. Cada ejemplo $\mathbf{x}^{(k)}$ se presenta con una probabilidad P_k que es proporcional al valor absoluto del error e_k; con lo cual se seleccionan con alta frecuencia las observaciones que tienen un error grande. A medida que se actualizan los pesos de la red, se actualizan también las probabilidades de presentación.

- **Repetición dependiendo del error ("Error-Dependent Repetition" -EDR):**

  En este caso las muestras se seleccionan de manera determinista como sigue:

  1. Se presenta el conjunto de entrenamiento completo.
  2. Se calcula el valor absoluto del error de las muestras del entrenamiento y su máximo $e_{\max}$.
  3. Durante un determinado número de épocas[1] W se hace una selección de muestras, escogiendo en la i-ésima época aquéllas cuyo error es superior a $i\, e_{\max}/W$.
  4. Se repiten los pasos desde 1 a 3 hasta alcanzar un criterio de parada.

- **Sistema de Carpetas ("Card-File System" -CFS):**

  Se hace una partición del conjunto de entrenamiento en N particiones. Sean $\{f_1, f_2, \ldots, f_N\}$ (con $f_1 < f_2 < \ldots < f_N$) las frecuencias de presentación de los ejemplos a la red, que se van asignar a cada partición. En cada época, se asigna una frecuencia de presentación a cada partición del modo que se da frecuencia más alta a particiones cuyas muestras suponen un error cuadrático medio mayor. El entrenamiento con dichas particiones se realiza de modo determinista o aleatorio (es decir, ponderando con $\{f_n\}$ o presentando las muestras con esas frecuencias).

- **Partición del conjunto de entrenamiento ("Training Set Partition" -PART):**

  Se divide el conjunto de entrenamiento en N partes, $S_1, S_2, \ldots, S_N$. El entrenamiento comienza con S_1 y después de un cierto número de épocas se añaden las muestras de S_2, etc. La presentación de muestras es uniforme. Hay dos formas de añadir el nuevo subconjunto S_i al conjunto de entrenamiento:

[1] El autor lo fijó en 100; se puede determinar por validación cruzada.

a) Entrenar exclusivamente con la parte nueva S_i, y luego continuar el entrenamiento con las muestras de $S_1 \cup S_2 \cup \ldots \cup S_i$.

b) Continuar inmediatamente el entrenamiento con las muestras de $S_1 \cup S_2 \cup \ldots \cup S_i$.

- **Repetir hasta aprender ("Repeat Until Learned" -RUL):**

  En realidad esta estrategia fue inventada por Munro [Munro1992], pero el parámetro β hace que las prestaciones dependan de las características del problema. Cachin adaptó el parámetro β para que su valor tenga la forma de $c\,E$, siendo E el error cuadrático medio para el conjunto de entrenamiento y c una constante (en [Cachin1994] c se fijó en 1.5 porque valores de c superiores a 1 resultan útiles). Así, se produce una repetición de las muestras que provocan un error mayor que el medio.

En cuanto a resultados, las estrategias de tipo determinista son más eficaces que las tipo aleatorio; estas últimas ofrecen malas prestaciones porque hay dependencia del problema de clasificación que se considere. Además, los experimentos de [Cachin1994] demuestran que el método EDR es ventajoso en comparación con las demás.

Validación Cruzada (CV, "Cross Validation") y GM

En la práctica, para realizar el entrenamiento de una red neuronal el conjunto de datos disponibles debe dividirse en tres:

- Conjunto de entrenamiento, para realizar la búsqueda que parametriza el modelo.
- Conjunto de validación, para observar cómo evoluciona la generalización.
- Conjunto de prueba, para medir las prestaciones de la máquina final.

Los dos primeros conjuntos constituyen el llamado conjunto de diseño.

La CV es un procedimiento de entrenamiento que permite obtener una buena generalización. El conjunto de diseño se divide en F partes: F-1 se usan para el entrenamiento y el subconjunto restante se reserva para la validación del modelo, es decir, medir sus prestaciones y seleccionar de acuerdo con ello valores para los parámetros no entrenables (dimensionales, por ejemplo). El procedimiento se repite F veces, cada vez con un conjunto de validación distinto.

A veces se dispone de un número escaso de datos; por lo tanto es arriesgado recurrir a la CV por dos motivos: primero, porque hace falta un número relativamente alto de datos, y segundo, por la pérdida de representatividad de la distribución de los subconjuntos de datos. En este caso se recurre al método "Leave One Out", en el cual cada vez se entrena el modelo con todo el conjunto de entrenamiento dejando fuera una muestra para la validación. El inconveniente de este método es su alto coste computacional.

En [Leisch1998] se propone un método de GM que determina el subconjunto del conjunto de entrenamiento sobre el que se aplica la CV. Este subconjunto contiene las muestras erróneas y las cercanas a la frontera de decisión, que son la más representativas dentro del conjunto de entrenamiento.

2.2.2. Diseño de redes RBF con GM

Las técnicas de GM sirven también para la configuración de la estructura de una red RBF; concretamente, para determinar los centroides. Generalmente, el diseño de una red RBF se enfrenta a un compromiso entre tres aspectos: la complejidad del problema, la complejidad de la arquitectura de la red y las prestaciones de la misma. Para un problema determinado la pregunta que se formula durante la construcción de una red RBF es: ¿Cuál es el número adecuado de centroides para obtener buenas prestaciones?

Una mala elección del número de centroides llevaría a una degradación en las prestaciones de la red. Lowe [Lowe1989] determinó heurísticamente un número crítico de centroides por encima del cual la generalización se deteriora.

[Moody1992, Niyogi1996] son trabajos que han estudiado el compromiso entre la complejidad y la capacidad de generalización de la red.

[Chang1993] construye progresivamente la red RBF utilizando como centroides las muestras localizadas en los bordes de la frontera de clasificación. Los pesos de la red son los únicos parámetros a ajustar. El procedimiento es como sigue:

1. Se construye una red RBF con un solo centroide. La determinación de dicho centroide se realiza por elección al azar de entre los obtenidos mediante k-medias sobre las muestras de entrenamiento de una de las clases elegida aleatoriamente.

2. Se entrena la red RBF con el conjunto de entrenamiento.

3. Se determinan las muestras erróneas mediante la comparación de los signos de las salidas del clasificador RBF y las etiquetas correspondientes a las muestras del conjunto de entrenamiento.

4. De entre estas muestras erróneas, se seleccionan las muestras cercanas a la frontera de decisión, que tienen una salida del clasificador RBF inferior a un umbral definido por el autor.

5. Se aplica el proceso de agrupamiento mediante k-medias a las muestras seleccionadas para cada clase y se obtienen nuevos centroides.

6. Se construye el nuevo clasificador añadiendo estos centroides a los existentes en la red, y se vuelve al paso 2 hasta alcanzar un criterio de parada.

Lyhyaoui et al. [Lyhyaoui1999] ofrecen una alternativa para construir clasificadores RBF, desarrollando un nuevo concepto de GM basado en agrupamiento para determinar los centroides críticos y también las muestras críticas. Se proponen dos modos de GM:

1. Actuando sobre el resultado de una cuantificación vectorial para determinar los centroides correspondientes a cada clase, y de entre estos centroides se seleccionan los que van a ser los centroides de la red RBF del siguiente modo:

se determinan los pares de los centroides de clases opuestas más cercanos entre sí como en [Sklansky1980] y se construye un conjunto de estos pares de centroides más cercanos, A_1 (como conjunto de centroides críticos); estos pares determinan directamente la frontera al tener que pasar ésta entre ellos. Si estos pares son suficientes para clasificar los otros centroides con kNN, no se seleccionan más centroides. Si no (en el caso en que resulten centroides mal clasificados), se construye un conjunto A_2 donde se incluye el centroide mal clasificado más próximo a un centroide crítico de la clase contraria. Se itera el proceso hasta que no haya centroides mal clasificados. El conjunto final de los centroides críticos es $A_1 \cup A_2$, cuyos elementos se utilizan como centroides de la red RBF final.

2. Añadiendo una selección local, escogiendo las muestras críticas pertenecientes a los agrupamientos representados por los centroides críticos como centroides de la RBF. Este modo de selección construye clasificadores RBF tipo SVM. Las muestras críticas se seleccionan como sigue: se consideran muestras correspondientes a cada centroide crítico y se eligen las más cercanas a la frontera de clasificación (proximidad a la frontera) y más típicas dentro de la distribución de la agrupación correspondiente al centroide que pertenecen (se asume que una muestra es más típica si está más cerca del punto correspondiente a la proyección de dicho centroide sobre la frontera de decisión en el espacio de características), para lo cual Lyhyaoui et al. usan una función indicadora (véase el detalle de esta función en [Lyhyaoui1999]) que incluye la proximidad a la frontera y la tipicidad.

Además, se utilizan los centroides críticos o las muestras críticas para preestimar las varianzas de las gaussianas de las RBFs; para los dos casos, las varianzas son proporcionales a la semidistancia d o d' multiplicada por un factor de proporcionalidad[2] δ, siendo d y d' la mitad de la distancia euclídea entre los dos centroides críticos de clases contrarias que están más cercanos entre sí y la mitad de la distancia euclídea entre las dos muestras críticas de clases contrarias que están más cercanas entre sí, respectivamente.

[2]En [Lyhyaoui1999] se comprueba que, para el caso de los centroides críticos, δ vale $\sqrt{2}$, y para el caso de las muestras críticas, que δ es igual a $3\sqrt{3/2}$.

2.2.3. Aplicación de GM para SVMs

Una serie de trabajos [Almeida2000, Shin2002, Shin2003a, Shin2003b, Shin2007] proponen la GM para mejorar el diseño de SVMs y reducir la complejidad computacional que requieren dichas máquinas cuando se dispone de bases de datos masivas.

Almeida et al. [Almeida2000] desarrollaron un procedimiento de GM, "SVM KM", basado en el agrupamiento k-medias para acelerar el entrenamiento de las SVMs. Se descartan agrupamientos formados solamente por muestras que pertenecen a la misma clase y se usan únicamente sus centroides para entrenar las SVMs, y se emplean además todas las muestras de los agrupamientos que contienen muestras de diferentes clases para dicho entrenamiento.

En [Shin2002] se propuso un método de GM también basado en el algoritmo kNN. El método selecciona las muestras cercanas a los bordes de decisión y bien clasificadas. Dichas muestras se emplean como vectores soporte. Se utilizan dos medidas, "proximidad" y "corrección":

- **Proximidad**: un patrón cerca de la frontera de decisión tiende a tener vecinos de diferentes clases. Una medida entrópica sobre los kNN sirve para medir la proximidad.
- **Corrección**: se define como la "probabilidad" de pertenencia a la clase correcta según el voto de los kNN.

Se utilizan muestras próximas a la frontera y suficientemente correctas para la arquitectura inicial de la SVM, descartando las demás.

El algoritmo es como sigue:

1. Encontrar los kNN a cada muestra $\mathbf{x}$.
2. Para cada muestra $\mathbf{x}$, calcular los votos de los kNN de las J clases

$$P_j(\mathbf{x}) = \frac{\sum_{i=1}^{k} 1 \quad \text{si} \quad F_i(\mathbf{x}) = j}{k}, \quad i = 1, \ldots, n, \qquad j = 1, \ldots, J. \tag{2.1}$$

$F_i(\mathbf{x})$ es la etiqueta del i-esimo vecino más próximo a $\mathbf{x}$, $F_i(.) \in \{1, \ldots, J\}$.

3. Calcular la proximidad de $\mathbf{x}$ a la frontera de decisión

$$\textbf{proximidad}(\mathbf{x}) = \sum_{j=1}^{J} P_j(\mathbf{x}) \log_J \frac{1}{P_j(\mathbf{x})} \tag{2.2}$$

(se toma $0 \log 0$ como 0).

4. Calcular la corrección de $\mathbf{x}$:

$$\textbf{corrección}(\mathbf{x}) = P_{j^*}(\mathbf{x}), \tag{2.3}$$

donde j^* es la etiqueta de la muestra $\mathbf{x}$.

5. Seleccionar los patrones que satisfacen
proximidad($\mathbf{x}$)> 0 y **corrección**($\mathbf{x}$) $\geq 1/J$.

Este método ofrece varias ventajas: en [Shin2002] se demuestra que el algoritmo anterior descarta las muestras mal clasificadas, con lo cual se convierte un problema no separable en un problema separable y así no es necesario buscar un valor óptimo del parámetro de regularización; y se reduce el tiempo de diseño de las SVMs sin perder la precisión en la tarea de clasificación. La dificultad del método radica en la determinación de k.

El mismo autor, en [Shin2003a], ha propuesto una variante del algoritmo anterior aplicando el procedimiento "Selective kNN Spanning" que reduce el coste computacional de ejecución. La idea consiste en aplicar el kNN sobre las muestras que están cerca de la frontera, y no a todo el conjunto de entrenamiento. Para completar ese algoritmo, Shin et al. [Shin2003b, Shin2007] han propuesto un método para determinar k, el número de vecinos. El procedimiento es del siguiente modo:

1. Aplicar 1 NN sobre el conjunto de entrenamiento.
2. Calcular la tasa de error de entrenamiento P_{error}.
3. Calcular $v = 2\,K\,P_{\text{error}}$, siendo K el tamaño del conjunto de entrenamiento.

4. Hallar el k^* óptimo como $k^* = \text{mín}\{k|b_k \geq v, k = 2, \ldots, K-1\}$, donde b_k es el número de muestras seleccionadas por el último algoritmo que hemos expuesto previamente en estas páginas.

Cabe destacar que la mayoría de los trabajos mencionados previamente han considerado dos aspectos esenciales para determinar las muestras relevantes:

- criterios de distancia, que permiten determinar las muestras cercanas a la frontera de clasificación;
- criterios de error, para determinar las muestras que generan un error grande.

2.2.4. Método de énfasis para Boosting

En [Gómez-Verdejo2006, Gómez-Verdejo2008] se ha propuesto un método de énfasis que presta atención a las muestras que producen un error grande y a las que están cerca de la frontera, con el fin de mejorar las prestaciones del diseño Real AdaBoost.

Sea $D_{n+1}(\mathbf{x}^{(k)})$ la función de énfasis del algoritmo Real AdaBoost, que se define en la forma:

$$D_{n+1}(\mathbf{x}^{(k)}) = D_n(\mathbf{x}^{(k)}) \frac{\exp\left(-\alpha_n\, o_n(\mathbf{x}^{(k)}) t^{(k)}\right)}{Z_n} \tag{2.4}$$

siendo $t^{(k)} = t(\mathbf{x}^{(k)})$, $o_n(\mathbf{x}^{(k)})$, α_n, y $Z_n = \sum_{k=1}^{K} D_n(\mathbf{x}^{(k)}) \exp\left(-\alpha_n o_n(\mathbf{x}^{(k)}) t^{(k)}\right)$, la etiqueta que corresponde a la muestra $\mathbf{x}^{(k)}$, la salida de la máquina base en la ronda n, el peso correspondiente a la máquina parcial n y el factor de normalización (que debe garantizar $\sum_{k=1}^{K} D_{n+1}(\mathbf{x}^{(k)}) = 1 \;\; \forall\, n$), respectivamente.

$D_{n+1}(\mathbf{x}^{(k)})$ se puede descomponer como sigue:

$$D_{n+1}(\mathbf{x}^{(k)}) = \frac{\exp\left(\frac{(f_n(\mathbf{x}^{(k)}) - t^{(k)})^2}{2}\right) \exp\left(-\frac{(f_n(\mathbf{x}^{(k)}))^2}{2}\right)}{\tilde{Z}_n} \tag{2.5}$$

donde $f_n(\mathbf{x}^{(k)})$ y $\tilde{Z}_n$ son la salida de la máquina parcial correspondiente a la ronda n y el factor de normalización, respectivamente. Se observa que $D_{n+1}(\mathbf{x}^{(k)})$ es la combinación fija de dos factores, representando cada uno de ellos un tipo diferente de énfasis:

1. Según el error cuadrático, correspondiente a $\exp\,(\frac{(f_n(\mathbf{x}^{(k)}) - t^{(k)})^2}{2})$, que es proporcional al error producido por cada muestra;

2. Según la proximidad a la frontera de clasificación, dada por $\exp\,(-\frac{(f_n(\mathbf{x}^{(k)}))^2}{2})$;

Gómez-Verdejo et al. definieron una nueva función de énfasis mixta mediante una combinación convexa entre los dos términos de énfasis, con la forma siguiente:

$$D_{\lambda,\,n+1}(\mathbf{x}^{(k)}) = \frac{1}{Z_{\lambda,\,n}} \exp\,[\lambda(f_n(\mathbf{x}^{(k)}) - t^{(k)})^2 + (1-\lambda) f_n(\mathbf{x}^{(k)})^2] \tag{2.6}$$

siendo λ $(0 \leq \lambda \leq 1)$ el parámetro de ponderación. $D_{\lambda,\,n+1}(\mathbf{x}^{(k)})$ permite controlar la atención concedida a los distintos tipos de datos según el valor del parámetro λ. Destacan tres casos particulares asociados con los valores de λ: los clasificadores centran su atención en las muestras críticas ($\lambda = 0$), en ambos términos de énfasis (función del Real AdaBoost estándar) ($\lambda = 0.5$), y en las muestras que generan un mayor error cuadrático ($\lambda = 1$).

En [García-Pedrajas2009] se desarrolló un método de Boosting basado en las técnicas de GM, en el que se construyen máquinas conjuntas como en AdaBoost usando como clasificadores base kNNs, SVMs y árboles de decisión basados en el algoritmo C4.5 [Quinlan1979]. La idea general del método es como sigue. En primer lugar, se entrena un primer clasificador base, de salida $o_0(\mathbf{x})$, con todas las muestras del conjunto de entrenamiento. Para entrenar el n-ésimo clasificador base, se obtiene la distribución de las ponderaciones como en AdaBoost, $D'_n(\mathbf{x})$. Después se obtiene un subconjunto S_n (a partir del conjunto original de entrenamiento) seleccionando muestras que presentan dificultades al aprendizaje y que minimizan el error ponderado[3] $\epsilon_n = E\{D'_n(\mathbf{x})_{[t(\mathbf{x}) \neq f_{n-1}(\mathbf{x})]}\}$, siendo $f_{n-1}(\mathbf{x})$ y $t(\mathbf{x})$

[3] $D'_{n+1}(\mathbf{x}) = D'_n(\mathbf{x})\beta_n^{1-[t(\mathbf{x})=o_n(\mathbf{x})]}$ con $\beta_n = \frac{\epsilon_n}{1-\epsilon_n}$.

la salida parcial del sistema en la $(n-1)$-ésima ronda[4] y la etiqueta correspondiente a la muestra $\mathbf{x}$, respectivamente. Finalmente, el n-ésimo clasificador base se entrena solamente con S_n.

La propuesta de García-Pedrajas pretende que cada clasificador del conjunto se entrene con muestras que sean relevantes para el proceso de aprendizaje, mejorando las prestaciones, y también reducir el conjunto de entrenamiento (reducir la complejidad del espacio de características) durante el entrenamiento. Los resultados experimentales muestran una mejora en las prestaciones del diseño propuesto con respecto a los clasificadores estándares kNN, SVM y C4.5.

En lo que sigue presentamos una alternativa a las técnicas de GM, que reemplaza las etiquetas duras por una versión blanda, aprovechando para introducir un énfasis que considere tanto el error como la proximidad a la frontera.

2.3. Blancos Blandos

2.3.1. Precedentes

Entre los trabajos que se interesaron por la idea de blancos blandos, cabe señalar [Reed1992, Reed1995], en los que se verificó que ablandar las etiquetas añadiendo ruido vía convolución ofrece buenos resultados. En [Mora-Jiménez2009] se propone una forma estática y otra iterativa de reducción selectiva de las etiquetas, SRTL ("Selective Reduction Target Level"), para mejorar las prestaciones de clasificadores RBFNNs. Consiste en reducir las etiquetas originales de acuerdo con la proximidad de las muestras a la frontera de decisión. El algoritmo estático aplica un clasificador auxiliar para estimar la proximidad de las muestras de entrenamiento al borde de decisión, y se usa la salida de la guía auxiliar para reducir la etiqueta correspondiente a cada muestra. Así obliga a que el proceso de entrenamiento preste atención a las muestras que presentan dificultades de aprendizaje. La versión iterativa del SRTL ofrece ventajas en varios problemas

[4] $f_{n-1}(\mathbf{x}) = \arg\max_{t(\mathbf{x})} \sum_{n'=1}^{n-1} \alpha_{n'}[o_{n'}(\mathbf{x}) = t(\mathbf{x})]$ con $\alpha_{n'} = \log \frac{1}{\beta_{n'}}$.

reales de clasificación con respecto a máquinas convencionales. Dichas ventajas se pueden resumir en la reducción del tiempo de cálculo durante el aprendizaje, la reducción del tamaño de los diseños y, finalmente, el aumento de la tasa de acierto.

2.3.2. Propuesta: Blancos Blandos Enfatizados (ESTs, "Emphasized Soft Targets")

Siguiendo con la idea de actuar sobre las etiquetas para enfatizar el entrenamiento con las muestras importantes, proponemos en esta Tesis un procedimiento de creación de blancos blandos que incluye énfasis.

Como se ha anticipado, se trata de crear ESTs mediante un procedimiento suficientemente general y flexible como para que resulte posible obtener buenas prestaciones renunciando a la inclusión de activaciones en la formulación transformada; en decir, aplicando directamente procedimientos de estimación (regresión). Por ello, se ha elegido construir los blancos blandos mediante una combinación convexa local de las etiquetas (duras) originales con las salidas de una guía (clasificador) auxiliar (que puede ser cualquier máquina de aprendizaje). A su vez, el parámetro de combinación lineal se hace depender del tamaño del error según una expresión multiparamétrica que permite tener en cuenta la proximidad a la frontera y el tamaño del error de manera muy flexible. Concretamente, las etiquetas blandas se construyen según:

$$t_s(\mathbf{x}) \;=\; \lambda(|e(\mathbf{x})|)\; t(\mathbf{x}) \;+\; (\,1 - \lambda(|e(\mathbf{x})|)\,)\; \mathrm{o}_{aux}(\mathbf{x}) \tag{2.7}$$

donde $\mathrm{o}_{aux}(\mathbf{x})$, $t(\mathbf{x})$, y $e(\mathbf{x})$ son la salida de la máquina auxiliar, la etiqueta original y el error de la máquina auxiliar correspondiente al dato $\mathbf{x}$, respectivamente; $\lambda(|e(\mathbf{x})|)$ es el peso de la combinación convexa que focaliza el entrenamiento de la segunda máquina (máquina final) sobre las muestras importantes (mal clasificadas y cercanas a la frontera de clasificación). El efecto del énfasis que se incluye en la etiqueta blanda sobre el comportamiento de la máquina final es tal que sólo las muestras críticas tienen una contribución importante durante el aprendizaje; las muestras bien clasificadas y los "outliers" intervienen poco al

aprendizaje de la máquina.

Para poder conseguir lo anterior, entre muchas posibilidades teóricas, una forma adecuada para el parámetro de ponderación $\lambda(|e(\mathbf{x})|)$ es

$$\lambda(|e(\mathbf{x})|) = \begin{cases} \exp(-\dfrac{(|e(\mathbf{x})| - \mu)^2}{\alpha_1}) & \text{para} \quad |e(\mathbf{x})| \leq \mu, \\ \\ \exp(-\dfrac{(|e(\mathbf{x})| - \mu)^2}{\alpha_2}) & \text{para} \quad \mu < |e(\mathbf{x})| \leq 2. \end{cases} \tag{2.8}$$

μ, α_1 y α_2 son los parámetros libres de las campanas de Gauss, que pueden ser determinados por CV, y $e(\mathbf{x})$ es el error de clasificación de la guía auxiliar. La ex-

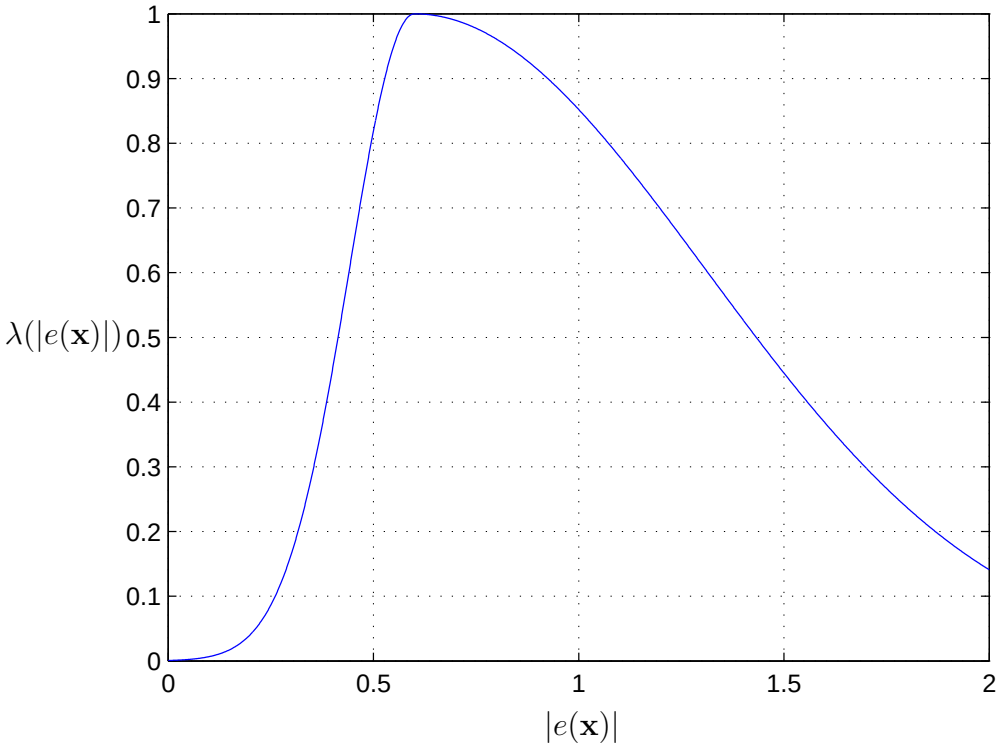

Figura 2.1: Forma de $\lambda(|e(\mathbf{x})|)$ para $\mu = 0.6$, $\alpha_1 = 0.05$ y $\alpha_2 = 1$.

presión (2.8) permite obtener el comportamiento deseado de la máquina final; su forma, compuesta por dos Gaussianas, concede importancia en el entrenamiento de la máquina final a las muestras según el valor producido por su error en la clasificación preliminar (véase la Fig. 2.1).

El parámetro μ determina el valor del error $|e(\mathbf{x})|$ donde la etiqueta blanda alcanza su máximo valor absoluto (la unidad), ya que $\lambda(|e(\mathbf{x})| = \mu) = 1$ y $t_s(\mathbf{x}) = t(\mathbf{x})$; y los parámetros α_1 y α_2 controlan el decaimiento a partir de este valor hacia las muestras con $|e(\mathbf{x})| \to 0$ (muestras bien clasificadas) o con $|e(\mathbf{x})| \to 2$ (muestras altamente erróneas). Se puede seleccionar el punto μ en el cual se aplica mayor énfasis, y se puede controlar independientemente el énfasis de las muestras con menor y con mayor error absoluto. Cuando $\mu = 1$, la ecuación (2.8) presta más atención a las muestras localizadas alrededor de la frontera que a las muestras erróneas; lo que es razonable si consideramos que se desea clasificar correctamente las muestras que no están lejos de la frontera de decisión, pero no vale la pena mover la frontera para atender muestras lejanas de la frontera y mal clasificadas: dicho esfuerzo provocaría una degradación en las prestaciones de la máquina. Por otra parte, si μ se acerca a 2, se focaliza el entrenamiento en las muestras altamente erróneas.

El mecanismo de énfasis sugerido actúa cualitativamente del siguiente modo:

- Cuando $|e(\mathbf{x})| \to 0$ (muestras bien clasificadas) o cuando $|e(\mathbf{x})| \to 2$ (muestras de alto error), $\lambda(|e(\mathbf{x})|)$ tiende a un valor pequeño, digamos ϵ; por lo tanto la ecuación (2.7) se transforma en:

$$t_s(\mathbf{x}) \approx \epsilon\ t(\mathbf{x}) + (1-\epsilon)\ \mathrm{o}_{aux}(\mathbf{x}) \ \approx\ \mathrm{o}_{aux}(\mathbf{x}). \tag{2.9}$$

  El valor de la nueva etiqueta es aproximadamente el valor de la salida de la guía auxiliar, y en el proceso de aprendizaje de la máquina final se acepta esa prestación.

- Si $|e(\mathbf{x})| \approx \mu$, el peso de ponderación $\lambda(|e(\mathbf{x})|)$ toma un valor λ_0 que se acerca a 1, de modo que

$$t_s(\mathbf{x}) \approx \lambda_0\ t(\mathbf{x}) + (1-\lambda_0)\ \mathrm{o}_{aux}(\mathbf{x}) \ \approx\ t(\mathbf{x}) \tag{2.10}$$

  lo que implica que la máquina final preste atención a las muestras que producen un error absoluto de valor μ (si μ se acerca al valor 1, estamos hablando de muestras cercanas a la frontera de decisión).

2.4. Conclusiones

Tras revisar trabajos relevantes de GM, se puede concluir que estas técnicas, a pesar de sus diferencias, tienen como objetivo identificar o seleccionar las muestras importantes para el aprendizaje con el fin de lograr una buena generalización. En la mayoría de los trabajos de GM mencionados anteriormente se consideran dos aspectos esenciales: la proximidad a la frontera y el error producido por las muestras.

A continuación, se ha presentado una propuesta de ESTs suficientemente general y flexible como para posibilitar que la aplicación directa de procedimientos de estimación (regresión) sin hacer uso de activaciones lleve a obtener buenas prestaciones desde el punto de vista del problema de decisión (clasificación). Los ESTs se obtienen mediante una combinación convexa local de las etiquetas duras y las salidas de una máquina auxiliar, dependiendo el coeficiente de combinación del tamaño del error de una forma adecuada y en la que se dispone de varios (tres) parámetros ajustables -que lo serán por CV en las aplicaciones prácticas-.

En los siguientes capítulos aplicaremos la idea de ESTs para diseñar nuevos clasificadores concebidos a partir de diferentes máquinas de aprendizaje.

Capítulo 3

Diseño de clasificadores MLPs basados en ESTs

3.1. Redes Neuronales

De manera general, situamos el origen de las redes neuronales en 1943 con el trabajo de McCulloch y Pitts [McCulloch1943], que presentaron el primer modelo neuronal artificial inspirado en el sistema nervioso humano. Este primer trabajo fue un punto de partida para desarrollar otros modelos neuronales. En 1949, Hebb [Hebb1949] definió dos conceptos muy importantes y fundamentales que han tenido una influencia en el campo de las redes neuronales, basándose en investigaciones psicofisiológicas: el pensamiento se desarrolla en el cerebro mediante conjuntos de neuronas activas simultáneamente, y la memoria (lo aprendido) se localiza en las conexiones entre las neuronas. La primera conferencia sobre Inteligencia Artificial en la que se discutió la capacidad de las computadoras para simular el aprendizaje fue en 1956 en Dartmouth (Estados Unidos). A partir de ahí muchos investigadores han desarrollado distintos tipos de redes neuronales. En 1959, Widrow [Widrow1959] publicó una teoría sobre la adaptación neuronal y unos modelos inspirados en esa teoría, como el Adaline ("Adaptive Linear Neuron") y el Madaline ("Multiple Adaline"); esos modelos se usaron en numerosas aplicaciones y permitieron emplear, por primera vez, una red neuronal en un problema real:

filtros adaptativos para eliminar ecos en las líneas telefónicas. [Rosenblatt1958] presentó el llamado "Perceptrón", una red formada por una sola neurona con activación dura y con pesos que se ajustan mediante la "Regla del Perceptrón" para resolver tareas de reconocimiento de patrones. En 1969, una sólida argumentación matemática [Minsky1969] demostró las limitaciones del Perceptrón con una sola capa para resolver problemas linealmente no separables, lo que provocó que la mayoría de los investigadores dejaran esa linea de investigación. A principios de los años 70, aparecieron trabajos sobre Mapas de Características Auto-Organizativos (SOFMs, "Self-Organizing Feature Maps") que utilizaban aprendizaje competitivo [Grossberg1972, Von der Malsburg1973, Grossberg1976a, Grossberg1976b]; posteriormente, estos estudios sirvieron a Kohonen para publicar un trabajo sobre SOFM [Kohonen1982]. Anderson et al. [Anderson1977] desarrollaron modelos de memorias asociativas y presentaron el asociador lineal conocido como el modelo "Brain-State-in-a-Box" (BSB). En 1982, Hopfield [Hopfield1982] propuso sus redes neuronales asociativas y el interés por las redes neuronales renació para los científicos. Mientras, Kohonen continuó el trabajo de Anderson y desarrolló redes de aprendizaje competitivo, una idea nueva basada en la biología: su principal aportación consiste en un procedimiento para conseguir que unidades físicamente adyacentes aprendieran a representar patrones de entradas similares; estas redes se denominan redes de Kohonen [Kohonen1984]. Después, [Rumelhart1986] publicó el algoritmo de retropropagación del error (BP,"BackPropagation"), que permite optimizar los parámetros de una red neuronal multicapa, aunque la idea básica de la BP fue de Werbos en [Werbos1974]; una formulación similar de la BP fue derivada por Parker [Parker1985], y el mismo algoritmo fue estudiado por LeCun [LeCun1985]. Desde entonces, han aparecido innumerables trabajos y aplicaciones comerciales de redes neuronales en diferentes ámbitos.

Dada la gran extensión de este campo, resultaría complicado hacer un estudio exhaustivo sobre las redes neuronales; por tanto, nos limitamos a describir brevemente los modelos neuronales y los métodos de aprendizaje para estructuras que se utilizarán en esta Tesis. Una amplia presentación general puede encontrarse en [Bishop2006, Haykin1999, Ripley1996].

Las redes neuronales se clasifican en dos tipos:

1. **Redes progresivas (FF, "FeedForward")**: en este tipo de arquitectura, el flujo de la información fluye en un solo sentido desde la capa de entrada hacia la salida a través de conexiones entre capas que incluyen pesos, sin existir ciclos ni conexiones entre neuronas de la misma capa. Una red de capa única consiste en una capa de entrada formada por un número determinado de nodos (la dimensión del vector de entrada) y una capa de salida que contiene una o varias neuronas (véase Fig. 3.1. a); como ejemplo podemos destacar el Perceptrón. Cuando dicha estructura se extiende a un mayor número de capas, se generan las redes multicapa. Dichas redes están compuestas por tres bloques: el primer bloque es la capa de entrada y el último bloque corresponde a la capa de salida; el bloque intermedio contiene una o varias capas ocultas (véase Fig. 3.1. b). Los Perceptrones MultiCapa

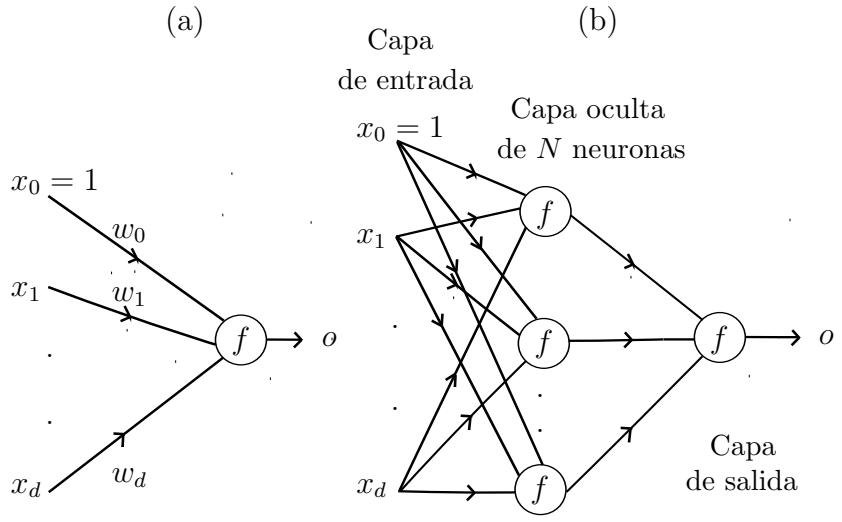

Figura 3.1: Representación esquemática de (a) una red de una capa compuesta de una sola neurona; $\mathbf{w}_e = [w_0, w_1, \ldots, w_d]^{\mathrm{T}}$ es el vector extendido de pesos; (b) una red multi-capa con una sola capa oculta formada por varias neuronas y una capa de salida compuesta por una sola neurona. $\mathbf{x}_e = [x_0, x_1, \ldots, x_d]^{\mathrm{T}}$, f, y o son el vector extendido de entrada, la activación, y la salida de una red, respectivamente.

(MLP, "MultiLayer Perceptrons"); las Redes de Funciones de Base Radiales (RBFNN, "Radial Basis Function Neural Networks") y las Redes Neuronales Probabilisticas (PNN, "Probabilistic Neural Networks") [Specht1990]

son ejemplos de redes neuronales pertenecientes a esta clase. Debe resaltarse que las FFNN tienen un comportamiento intrínsecamente estático: la salida depende únicamente de la entrada, y no de la evolución temporal. Si se desea tratar problemas espaciotemporales (con salida dependiente de la entrada y de la evolución temporal) he de mostrarse la evolución temporal a la entrada de la red; así ocurre con las redes neuronales con retardos temporales (TDNN, "Time Delay Neural Networks")[Waibel1987, Lang1990].

El resultado de la actividad de una neurona constituyente consiste en una suma ponderada de las entradas $\mathbf{x} = [x_1, x_2, \ldots, x_d]^{\mathrm{T}}$ (d es la dimensión del problema) seguida de la aplicación de una función no lineal f denominada activación, como se ilustra en Fig. 3.2. Se puede emplear diversas activaciones; por ejemplo, un escalón unitario (o signo), una sigmoide o una tangente hiperbólica, o una función rampa.

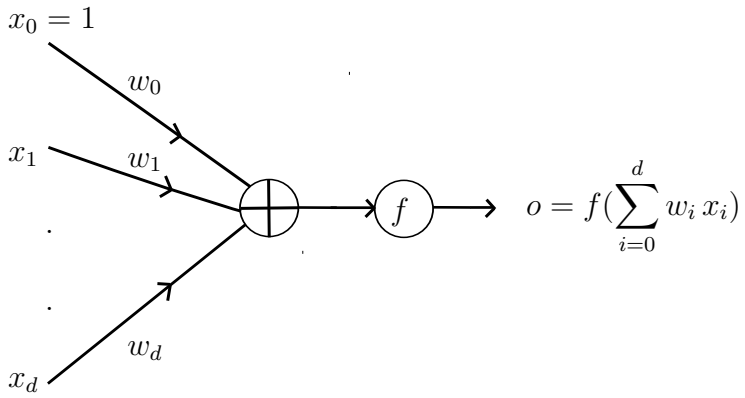

Figura 3.2: Esquema de una neurona artificial. $\mathbf{w}_e = [w_0, w_1, \ldots, w_d]^{\mathrm{T}}$, f y o son el vector extendido de pesos, la activación y la salida de la neurona, respectivamente.

2. **Redes recurrentes (RNN, "Recurrent Neural Networks")**: a diferencia de las redes progresivas, las redes recurrentes permiten la interconexión de neuronas en cualquier sentido y la inclusión de operadores de retardo. Generalmente, las capas de una red recurrente consisten en conjuntos de neuronas que reciben las salidas de las neuronas de las capas anteriores, junto con las salidas de la misma capa o de capas posteriores. La presencia de estas realimentaciones influye notablemente en la capacidad de aprendizaje: se puede aplicar a problemas espaciotemporales. De entre estas redes

destacan las redes de Hopfield [Hopfield1982] y los MLPs realimentados (redes de Elman [Elman1990], BP Through Time [Rumelhart1986], etc.). No las estudiaremos aquí por estar fuera del objetivo de esta Tesis.

El aprendizaje de una red neuronal es la forma en la que la red, durante el proceso de entrenamiento, accede a y asimila información sobre el comportamiento deseado. Haykin [Haykin1999] propone una clasificación del aprendizaje en tres tipos:

1. **Aprendizaje supervisado**: para cada patrón de entrada $\mathbf{x}^{(k)}$ disponemos de una etiqueta para la salida de la red, dando lugar al denominado conjunto de entrenamiento $\{(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})\}$, $k = 1, \ldots, K$. En este caso es posible construir un criterio de error en función de los valores deseados.

2. **Aprendizaje por refuerzo**: en este tipo de aprendizaje el entrenamiento está dirigido por un "crítico" mediante un mecanismo de penalización y recompensa, por el cual la red es recompensada por una salida correcta y castigada por una equivocada.

3. **Aprendizaje no supervisado**: en este último tipo de aprendizaje los patrones de entrada no disponen de las salidas deseadas para guiar el aprendizaje; por tanto, la red se diseña para conseguir un comportamiento adecuado bajo reglas generales. Es, por ejemplo, el caso de los SOFMs.

3.2. El Perceptrón MultiCapa

Como se ha dicho, un MLP es una red multicapa tipo FF que consiste en una capa de entrada de nodos sensores que reciben la información, una o varias capas ocultas y una capa de salida.

El entrenamiento supervisado de un MLP se realiza habitualmente con el algoritmo BP, que se basa en la regla de corrección del error empleando el método de gradiente [Widrow1959]; dicha regla consiste en minimizar un error (comúnmente cuadrático) por medio de gradiente descendiente, por lo que la parte esencial del

algoritmo es el cálculo de las derivadas parciales de dicho error con respecto a los parámetros de la red neuronal. Este algoritmo se aplica en dos etapas: en la primera se presenta cada muestra del conjunto de entrenamiento y esa entrada se propaga hacia adelante para calcular la salida; en esta etapa, los pesos no se modifican. La segunda etapa es hacia atrás, actualizando los pesos para acercar la salida al valor deseado.

Hay dos modos de presentación de los ejemplos: modo secuencial, muestra a muestra, u "online", y modo bloque, o "batch". En el primero, los pesos se actualizan después de la presentación a la red de cada ejemplo. En el segundo, los pesos se modifican después de la presentación de todo el conjunto de entrenamiento. Se resume el algoritmo BP en el Apéndice A.

El algoritmo BP es un método de aprendizaje general cuya aplicación es sencilla. Pero, además, si se pretende afinar y obtener buenas soluciones hay que ser cuidadoso con la determinación de una arquitectura óptima (dimensionado), la inicialización de los pesos, los parámetros de aprendizaje (tasa de aprendizaje, momento,...), y utilizar técnicas para evitar el sobreajuste. No obstante su sencillez, en la aplicación del algoritmo BP pueden aparecer, entre otras, las siguientes dificultades:

- Parálisis de la red: los pesos toman valores grandes, y no se actualizan por efecto de la saturación de f.
- Mínimos locales: la superficie de error está llena de valles y picos. Es difícil localizar el mínimo global, y se suele caer en mínimos locales.
- Lentitud del aprendizaje: si la tasa de aprendizaje es muy pequeña el algoritmo es muy lento, y si es grande pueden aparecer efectos oscilatorios o incluso divergencia.

Naturalmente, es preciso combatir estos problemas. Hay varios trabajos que discuten opciones para evitarlos [Rumelhart1986, Hopfield1987, Falhman1988, Vogl1988, Burrascano1991, Darken1992, Cichocki1993].

3.3. Aplicación de los ESTs a MLPs

En lo que sigue consideramos problemas de clasificación binaria.

En primer lugar, entrenamos una máquina auxiliar tipo MLP, MLP_{aux}, con el conjunto de entrenamiento original $\{(\mathbf{x}^{(k)}, t(\mathbf{x}^{(k)}))\}_{k=1}^{K}$. Indicamos la salida de dicha red como $\text{o}_{aux}(\mathbf{x})$. Después, utilizamos $\text{o}_{aux}(\mathbf{x})$ para calcular la etiqueta blanda $t_s(\mathbf{x})$ según (2.7) y (2.8) definidas en el segundo capítulo. La máquina final, $\text{EST-MLP}_{\text{MLP}}$, se entrena con el nuevo conjunto de entrenamiento $\{(\mathbf{x}^{(k)}, t_s(\mathbf{x}^{(k)}))\}$. En realidad, la máquina final estima el valor de la etiqueta blanda, y luego se aplica un decisor duro para asignar cada nueva muestra $\mathbf{x}^*$ a su correspondiente clase.

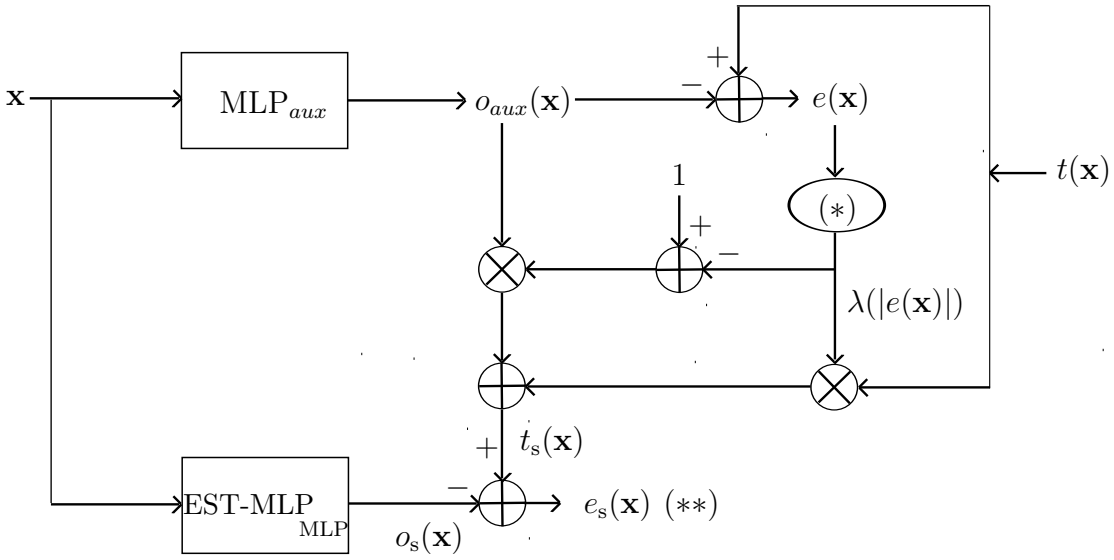

Figura 3.3: Esquema del clasificador $\text{EST-MLP}_{\text{MLP}}$ basado en ESTs. MLP_{aux} y $\text{EST-MLP}_{\text{MLP}}$ son la máquina auxiliar y final, respectivamente. (*) se refiere a la ecuación (3.2) para $\lambda(|e(\mathbf{x})|)$; (**), al control (no representado) de $\text{EST-MLP}_{\text{MLP}}$ con dicho error durante el entrenamiento.

Así, pues, tenemos el siguiente proceso (véase la Fig. 3.3):

1. Entrenar una máquina auxiliar, MLP_{aux}, con el conjunto de entrenamiento $\{(\mathbf{x}^{(k)}, t(\mathbf{x}^{(k)}))\}_{k=1}^{K}$.

2. Calcular la salida de la máquina auxiliar (previamente entrenada) $\mathrm{o}_{aux}(\mathbf{x}^{(k)})$ y el error $e(\mathbf{x}^{(k)})$ para la k-ésima muestra $\mathbf{x}^{(k)}$.

3. Determinar $t_s(\mathbf{x}^{(k)})$ según las expresiones (2.7) y (2.8), que se repiten aquí:

$$t_s(\mathbf{x}^{(k)}) \;=\; \lambda(|e(\mathbf{x}^{(k)})|)\; t(\mathbf{x}^{(k)}) \;+\; (\,1 - \lambda(|e(\mathbf{x}^{(k)})|)\,)\; \mathrm{o}_{aux}(\mathbf{x}^{(k)}) \tag{3.1}$$

con

$$\lambda(|e(\mathbf{x}^{(k)})|) = \begin{cases} \exp(-\dfrac{(|e(\mathbf{x}^{(k)})| - \mu)^2}{\alpha_1}) & \text{para} \quad |e(\mathbf{x}^{(k)})| \leq \mu, \\ \\ \exp(-\dfrac{(|e(\mathbf{x}^{(k)})| - \mu)^2}{\alpha_2}) & \text{para} \quad \mu < |e(\mathbf{x}^{(k)})| \leq 2. \end{cases} \tag{3.2}$$

siendo μ, α_1 y α_2 los parámetros libres de las campanas de Gauss, que se determinan por CV.

4. Entrenar la máquina final EST-MLP$_{\mathrm{MLP}}$ con $\{(\mathbf{x}^{(k)}, t_s(\mathbf{x}^{(k)}))\}_{k=1}^{K}$.

A la presentación de un nuevo dato $\mathbf{x}^*$, la decisión se realiza mediante un decisor duro

$$sgn(o_s(\mathbf{x}^*)) \underset{-1}{\overset{+1}{\gtrless}} 0 \tag{3.3}$$

siendo $o_s(\mathbf{x})$ la salida del EST-MLP$_{\mathrm{MLP}}$.

3.4. Pruebas Experimentales

3.4.1. Conjuntos de datos

Trabajaremos con tres conjuntos de datos frecuentemente utilizados en los problemas de clasificación. El primer problema es Ripley, que se ha sido utilizado en [Ripley1994, Ripley1996, Osuna1997, Cherkassky1998]. La tasa de error del clasificador bayesiano MAP es 8 %. Los motivos por los que estos datos se emplean como referencia son fundamentalmente dos: por una parte, el alto grado de solapamiento que presentan las dos clases, y por otra, el reducido número de

datos de entrenamiento con respecto al número de datos de prueba. La distribución de los datos obedece a una mezcla de Gaussianas (dos esféricas por clase) con los siguientes parámetros: los datos de la clase 1 tienen los centros en (-0.3, 0.7) y (0.4, 0.7), los de la clase -1 tienen los centros en (-0.7, 0.3) y (0.3, 0.3), y todas las Gaussianas tienen la misma varianza, 0.03. Los datos son accesibles públicamente en http://www.stats.ox.ac.uk/pub/PRNN/

La Fig .3.4 muestra los datos del conjunto de entrenamiento de Ripley ("+": clase 1; "o": clase -1)

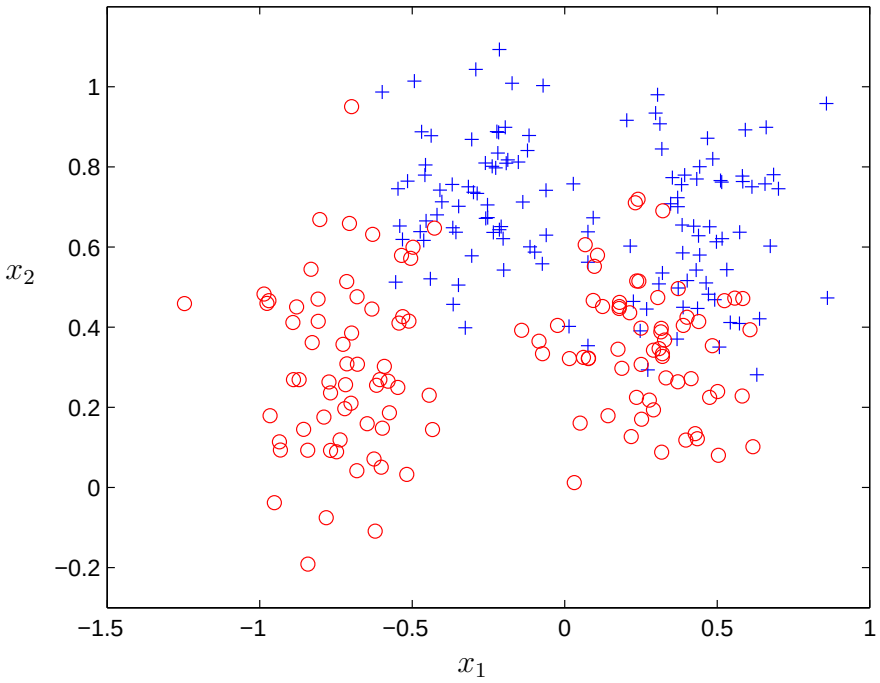

Figura 3.4: Representación del conjunto de entrenamiento de Ripley (de 125 muestras por clase).

El segundo problema es Ionosfera: corresponde a datos de un sistema radar recogidos en Goose Bay, Labrador (Canadá). Este sistema consiste en 16 antenas de alta frecuencia, en fase, con una energía transmitida total del orden de 6.4 kilowatios [Sigillito1989]. Los blancos son electrones libres en la ionosfera. Los retornos "Good" (clase 1) son los que muestran la detección de un blanco en la

ionosfera, mientras los "Bad" (clase -1) son los que no lo hacen. Los datos son accesibles públicamente en http://archive.ics.uci.edu/ml/datasets/Ionosphere

El tercer problema es Tic-tac-toe, una base de datos del juego "Tres en Raya" que representa una codificación del sistema completo de configuraciones posibles. El triunfo para el jugador "x" (se asume que el jugador "x" empieza primero) corresponde a la clase 1; la alternativa es el triunfo para el jugador "o" (clase -1) [Matheus1989, Matheus1990, Aha1991]. La base de datos es accesible públicamente en http://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame

Las principales características de los problemas figuran en la Tabla 3.1.

Problemas	Ionosfera	Ripley	Tictactoe
Dimensión	34	2	9
Entrenamiento	201	250	575
Clase +1	101	125	199
Clase −1	100	125	376
Prueba	150	1000	383
Clase +1	124	500	133
Clase −1	26	500	250

Tabla 3.1: Características de los tres problemas: Ionosfera, Ripley y Tictactoe.

3.4.2. Descripción de las simulaciones

Entrenamos las dos máquinas, la auxiliar MLP_{aux} y la final $\text{EST-MLP}_{\text{MLP}}$, con el algoritmo BP, utilizando el error cuadrático medio como función de coste, con tasa de aprendizaje de 10^{-3} y aplicando época por época el criterio de parada "Early Stopping" (ES). ES es una técnica simple de obtener una buena generalización que se utiliza cuando una máquina se entrena "online" por el gradiente descendiente, para evitar el sobreajuste. En ES, el conjunto de entrenamiento original se divide en un nuevo conjunto de entrenamiento y un conjunto de validación; en nuestros problemas, reservamos el 80 % para entrenamiento y el 20 %

para validación, eligiendo según una partición al azar en cinco bloques. Después de entrenar el modelo, la red se evalúa con el conjunto de validación. El algoritmo de entrenamiento se detiene cuando las prestaciones de la máquina sobre los datos de validación empiezan a deteriorar. La red con la mejor prestación con el conjunto de validación se utiliza para las pruebas con un conjunto distinto de datos (conjunto de prueba o "test"). En nuestro caso, durante el entrenamiento de un MLP, el algoritmo BP se detiene cuando el MSE del conjunto de validación empieza a crecer, y se eligen los pesos que corresponden al valor mínimo del MSE sobre el conjunto de validación. El número máximo de épocas se fija en 800, para garantizar la convergencia del entrenamiento. El entrenamiento de las máquinas MLP_{aux}, $\text{EST-MLP}_{\text{MLP}}$, y un MLP estándar que vamos a utlizar como referencia para comparaciones, se realiza mediante CV de 5 particiones del conjunto de entrenamiento original (80 % para entrenamiento y 20 % para validación); también, los parámetros libres de los diseños se seleccionan por la misma CV. Se realizan 10 inicializaciones (repeticiones) independientes del MLP por cada partición de la CV. Los parámetros libres de nuestro diseño se exploran como sigue:

- Número de neuronas de la capa oculta de la máquina de comparación (MLP convencional), de la auxiliar, y de la máquina final, N, N_{aux}, N_{ST} : {4, 6, 8, 10, 12, 14, 16}.
- μ: {0.1, 0.3, 0.6, 1, 1.2, 1.6, 2}.
- α_1, α_2: $\{10^{-3}, 10^{-2}, 5.10^{-2}, 0.1, 0.5, 1, 1.5, 2, 3, 4, 5\}$.

Los pesos de la máquina auxiliar MLP_{aux} y el MLP estándar se inicializan en cada repetición según una distribución uniforme en [-0.1, 0.1], y los pesos de la máquina final $\text{EST-MLP}_{\text{MLP}}$ se inicializan con los valores finales de los pesos de la máquina auxiliar (de acuerdo con dicha inicialización, condiciono que los pesos del $\text{EST-MLP}_{\text{MLP}}$ tengan la misma dimensión que los del MLP_{aux}).

También consideramos, como referencia, el método de selección de muestras EDR ("Error Dependent Repetition"; véase Capítulo 2) de Cachin. EDR utiliza un MLP de N' neuronas ocultas (N' se determina por CV en el mismo intervalo que N, N_{aux} y N_{ST}), que se entrena de la misma manera que el MLP estándar.

En la parte experimental, se ha probado con dos activaciones a la salida del EST-MLP$_{\text{MLP}}$, lineal y tangente hiperbólica, para estimar la etiqueta blanda $t_s(\mathbf{x})$. Teóricamente, para un problema de estimación usando los MLPs es lógico emplear una activación lineal a la salida. En nuestro caso, el uso de los ESTs permite el paso a activación lineal, pero si nos fijamos bien en la formulación de la ecuación (3.1), se nota que $t_s(\mathbf{x})$ está compuesta de una componente binaria, $t(\mathbf{x})$ y otra real, $o_{aux}(\mathbf{x})$. Por consiguiente, esto conduce a que el uso de la tangente hiperbólica sea adecuado si el efecto del parámetro de combinación convexa $\lambda(|e(\mathbf{x})|)$ da mucha importancia a la componente binaria. En caso contrario, si a la salida de la máquina auxiliar se le da más peso, sería recomendable usar la salida lineal. Por ello se hacen pruebas con las dos opciones y se elige la que ofrece mejores resultados (en CV).

3.4.3. Resultados

La Tabla 3.2 presenta la tasa de acierto del EST-MLP$_{\text{MLP}}$, que es el promedio de tasa de acierto de clasificación de 50 realizaciones independientes, (APCC, "Average Percentage of Correct Classification") en %, comparado con el del MLP estándar y el EDR; también incluimos los resultados de la aproximación "omnisciente": la que corresponde al diseño que tiene la mejor tasa de acierto de clasificación sobre datos de test; es verdad que la aproximación omnisciente no es un diseño lícito, pero sirve para ver las limitaciones del proceso de CV y las capacidades máximas de los diseños. En la Tabla 3.2, presentamos el mejor resultado según la activación seleccionada para la salida del EST-MLP$_{\text{MLP}}$: para Ripley, la tangente hiperbólica, y para Ionosfera y Tictactoe, la activación lineal: para el problema Ripley, el EST-MLP$_{\text{MLP}}$ con tangente hiperbólica como salida ofrece una pequeña ventaja respecto al EST-MLP$_{\text{MLP}}$ con salida lineal, mientras que en el caso de los datos reales el EST-MLP$_{\text{MLP}}$ con salida lineal ofrece la mejor tasa de acierto de clasificación superando ligeramente al EST-MLP$_{\text{MLP}}$ con tangente hiperbólica.

Nótese que, tal y como se pretendía, la aplicación de nuestro potente y flexible método de creación ESTs lleva a que incluir o no activaciones resulte poco

Problemas	Ionosfera	Ripley	Tictactoe
MLP CV	91.00±3.40	90.35±0.58	68.56±4.51
N	4	14	14
MLP omni*	91.34±4.07	90.49±0.44	69.84±5.63
N	8	10	16
EST-MLP$_{\text{MLP}}$ CV	92.12±2.66	90.60±0.34	71.69±4.40
$N_{aux}/N_{\text{ST}}/\mu/\alpha_1/\alpha_2$	14/4/1.6/2/1.5	12/8/1.6/0.05/4	14/4/1.6/4/3
EST-MLP$_{\text{MLP}}$ omni*	93.04±2.50	90.71±0.45	74.50±5.00
$N_{aux}/N_{\text{ST}}/\mu/\alpha_1/\alpha_2$	14/8/2/0.1/5	12/8/1.6/0.05/4	14/4/1.6/4/3
EDR CV	91.73±4.17	90.25±0.40	71.38±4.59
N'	12	14	8
EDR omni*	93.16±3.49	90.25±0.40	71.38±4.59
N'	16	14	8

Tabla 3.2: Tasa de acierto de clasificación sobre los datos de test (desviación estándar) con las máquinas MLP, EST-MLP$_{\text{MLP}}$ y EDR, para los tres problemas. "omni" se refiere a la aproximación "omnisciente".

relevante -posibilitando, por tanto, recurrir directamente a procesos de estimación (regresión)-, al tiempo que, como veremos, obtienen buenas prestaciones en clasificación (potencialmente superiores a los (más sencillos) métodos habituales de GM incluso en problemas de complejidad moderada como los aquí considerados). Esto abre el camino a aplicar estas ideas sobre otras familias de máquinas de aprendizaje, como haremos en capítulos posteriores.

En comparación con el MLP estándar, aparece una ligera ventaja del método EST en el problema Ripley, y una ventaja mayor en Ionosfera. La ventaja es aún más clara en Tictactoe. Además, se verifica que los resultados del EDR, aún acercándose más a los del EST-MLP$_{\text{MLP}}$, son algo inferiores. Esto es revelador del potencial de los ESTs. Cabe esperar que la aplicación del método a esquemas más elaborados conduzca también a mejores resultados.

Obviamente, hemos obtenido mejoras a costa de un alto esfuerzo de entrenamiento: necesitamos 7×7×7×11×11 diseños para cada guía: para el entrenamiento de nuestro diseño, se necesita explorar por CV los valores de los parámetros

libres, cuyo número es $\#N_{aux} \times \#N_{\mathrm{ST}} \times \#\mu \times \#\alpha_1 \times \#\alpha_2$, aunque el número de épocas se reduce (entre 25-50 %) porque la inicialización de los pesos del EST-MLP$_{\mathrm{MLP}}$ se realiza con los pesos finales del MLP$_{aux}$.

La Figura 3.5 ilustra la frontera de clasificación del EST-MLP$_{\mathrm{MLP}}$ (curva discontinua), del MLP estándar (curva continua gruesa), y del clasificador bayesiano (frontera teórica; curva continua fina) para el problema Ripley; se observa que la

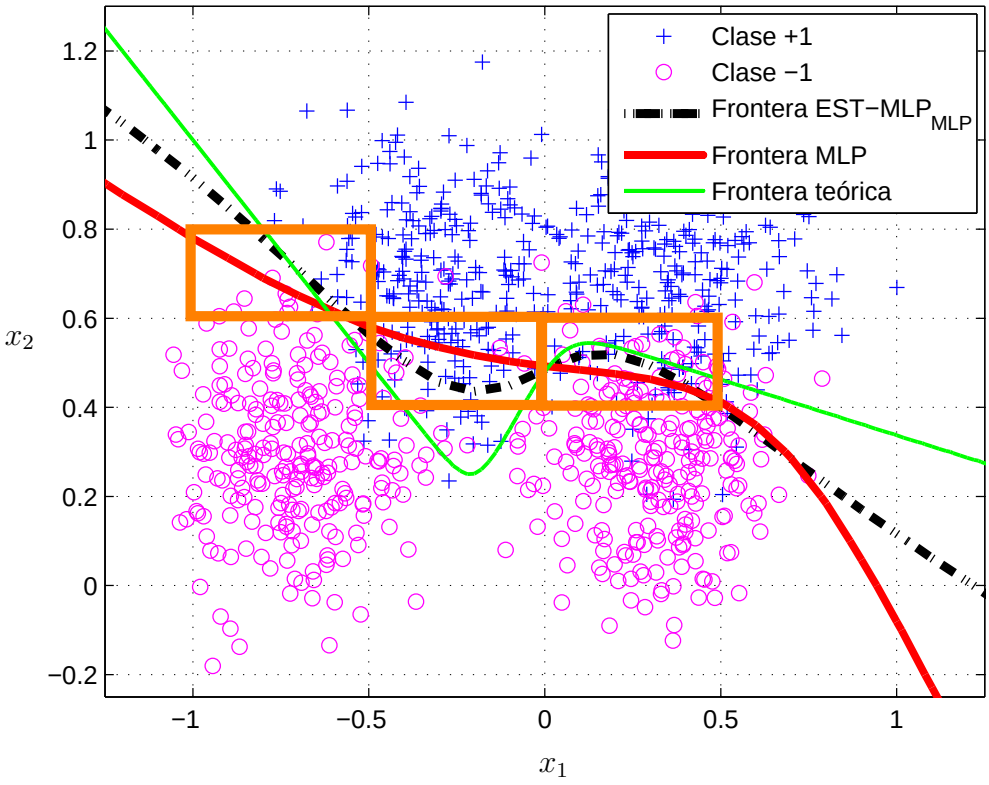

Figura 3.5: Fronteras de decisión sobre los datos de test para el problema bidimensional de Ripley (500 muestras para cada clase).

frontera de decisión del EST-MLP$_{\mathrm{MLP}}$ se ajusta mejor a la frontera teórica que la del MLP estándar porque el énfasis implícito en EST permite que la máquina final EST-MLP$_{\mathrm{MLP}}$ preste más atención en las muestras cercanas a la frontera y que el MLP convencional clasifica erróneamente; eso se ve claramente en las siguientes regiones determinadas por la coordenada x_1 en el espacio de entrada

(véase Fig. 3.5): región 1 ($-1 \leq x_1 \leq -0.5$), región 2 (-0.5$\leq x_1 \leq 0$) y región 3 ($0 \leq x_1 \leq 0.5$). Se percibe que el EST-MLP$_{\text{MLP}}$ recupera algunas muestras mal clasificadas por el MLP estándar de la clase -1 en las regiones 1 y 3 y, sobre todo, de la clase $+1$ en la región 2, y de ahí el aumento de la tasa de acierto del EST-MLP$_{\text{MLP}}$ con respecto al MLP estándar.

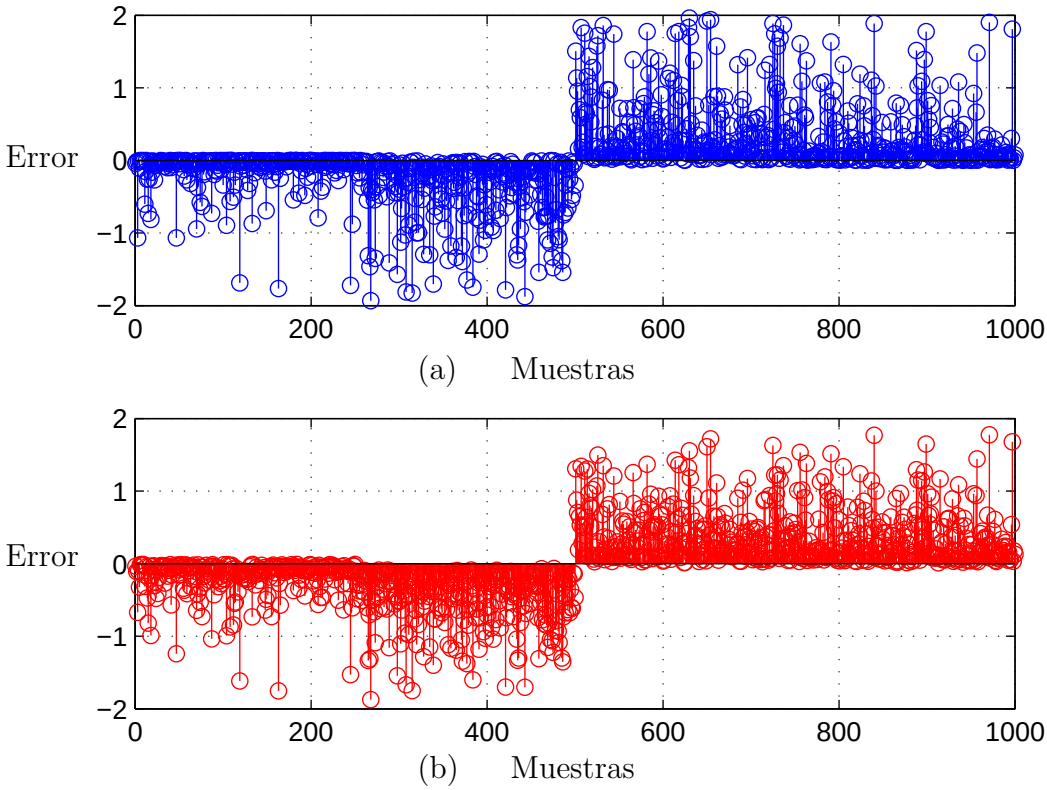

Figura 3.6: Error de test del MLP estándar (a) y del EST-MLP$_{\text{MLP}}$ (b) para el problema Ripley.

La Figura 3.6 representa el error sobre el conjunto de test del problema Ripley de las máquinas MLP y EST-MLP$_{\text{MLP}}$. Se percibe que hay una reducción notable del error producido por las muestras de test del EST-MLP$_{\text{MLP}}$ con respecto al MLP estándar; se observa esa reducción del error tras la comparación visual de (a) y (b) de la Figura 3.6 para las muestras posicionadas entre la 300-ésima y 800-ésima posición y que tienen el valor absoluto del error superior a 1; además,

el error cuadrático medio correspondiente las sub-figuras (a) y (b) de la Fig. 3.6 vale 0.3129 y 0.2933, respectivamente. Cuando aplicamos la primera máquina, hay varias muestras con valores importantes de error, y los pesos de la máquina no tienen la capacidad de reducir estos errores, dando lugar a muestras mal clasificadas. Pero si utilizamos la máquina EST-MLP$_{\mathrm{MLP}}$, el mecanismo de énfasis permite el ajuste de los pesos principalmente en función de las muestras relevantes; como consecuencia, es posible que la frontera se mueva hacia la posición de las muestras relevantes para recuperar éstas. Por lo tanto, el error cuadrático baja y el número de muestras mal clasificadas decrece.

Para estudiar la sensibilidad del EST-MLP$_{\mathrm{MLP}}$ de una manera sencilla, para evaluar las limitaciones del diseño CV (especialmente, cuando el diseño tiene varios parámetros libres a explorar), mediante la comparación entre la tasa de acierto del diseño encontrado por CV y el diseño correspondiente a la aproximación "omnisciente" (el límite superior del resultado de la CV), en que los parámetros se seleccionan teniendo en cuenta los resultados en las muestras de test (lo que no es un diseño admisible). Si los valores de la tasa de acierto y de los parámetros de diseño determinados por el proceso de CV se acercan a los del "omnisciente", se puede decir que el diseño presenta poca sensibilidad respecto a los valores seleccionados por CV, y, por tanto, el proceso es razonablemente eficaz.

De acuerdo con la Tabla 3.2, la diferencia entre el diseño CV y el diseño omnisciente para EST-MLP$_{\mathrm{MLP}}$ es relevante en Tictactoe -tasa de acierto- y en Ripley -parámetros de diseño-; en todo caso, en estos problemas se consigue ventaja aunque la CV no sea tan eficaz como cabría desear.

Finalmente, para ampliar la discusión sobre la sensibilidad, estudiamos la variación de la tasa de acierto de clasificación con los parámetros de diseño alrededor de los valores encontrados por CV. Las Figuras 3.7, 3.8, 3.9, 3.10 y 3.11 ilustran la sensibilidad respecto a los parámetros N_{aux}, N_{ST}, μ, α_1, y α_2 alrededor del punto que corresponde al diseño CV para los datos de test de los tres problemas: Ionosfera, Ripley y Tictactoe.

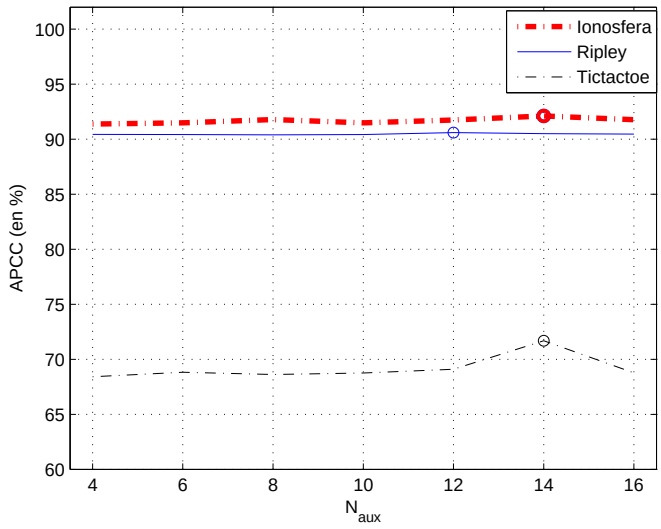

Figura 3.7: Sensibilidad respecto a N_{aux} del EST-MLP$_{\text{MLP}}$ sobre los datos de test de Ionosfera, Ripley y Tictactoe. Los círculos representan los valores encontrados por CV.

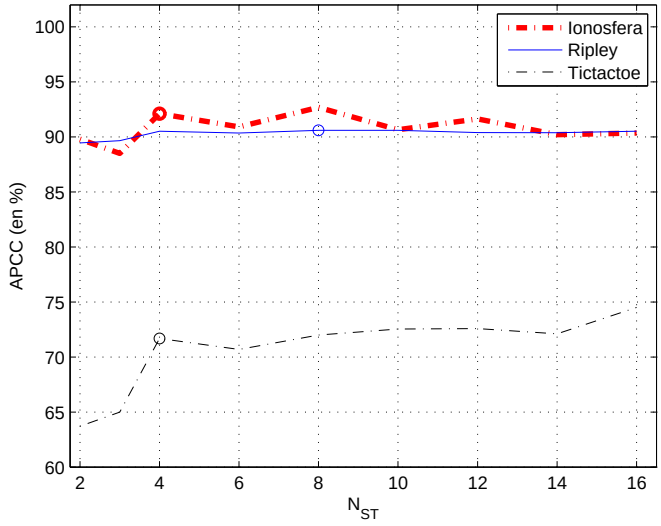

Figura 3.8: Sensibilidad respecto a N_{ST} del EST-MLP$_{\text{MLP}}$ sobre los datos de test de Ionosfera, Ripley y Tictactoe.

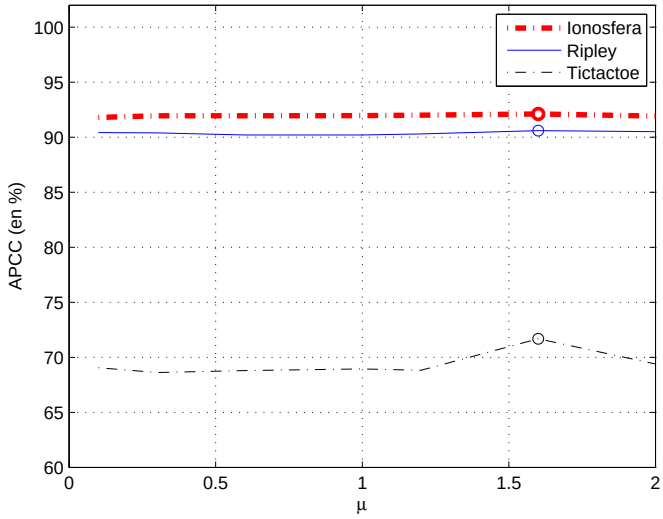

Figura 3.9: Sensibilidad respecto a μ del EST-MLP$_{\text{MLP}}$ sobre los datos de test de Ionosfera, Ripley y Tictactoe.

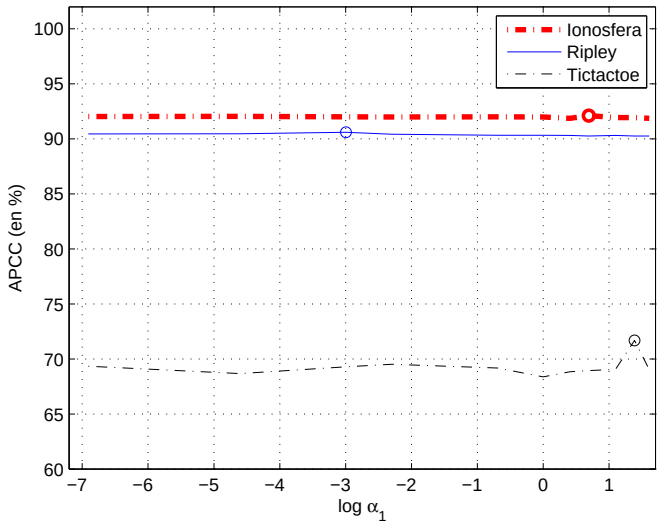

Figura 3.10: Sensibilidad respecto a α_1 del EST-MLP$_{\text{MLP}}$ sobre los datos de test de Ionosfera, Ripley y Tictactoe.

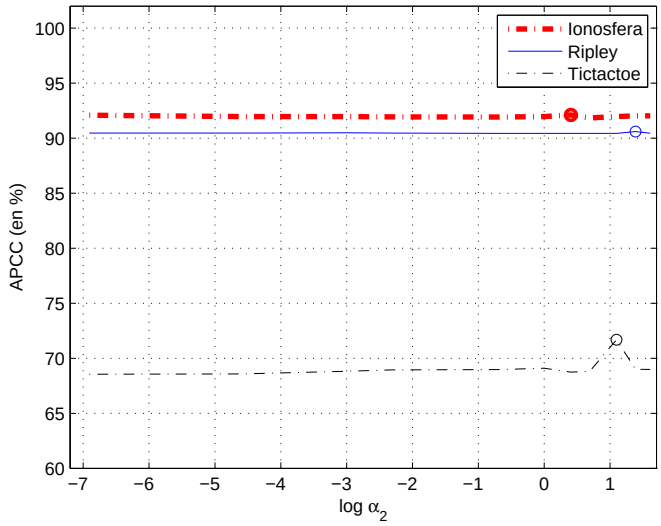

Figura 3.11: Sensibilidad respecto a α_2 del EST-MLP$_{\text{MLP}}$ sobre los datos de test de Ionosfera, Ripley y Tictactoe.

Para los datos Ionosfera y Ripley, las curvas de los parámetros N_{aux}, μ, α_1, y α_2 son planas; lo que significa que el diseño EST-MLP$_{\text{MLP}}$ es poco sensible respecto a estos parámetros. Mientras, para Tictactoe se nota que hay un pico alrededor del punto de CV (que se encuentra en zona de altas prestaciones), pero en las regiones lejos de este punto, los valores de la tasa de acierto son estables; se puede decir que hay sensibilidad limitada. Por otro lado, en las curvas correspondientes al parámetro N_{ST} se observa que existe una sensibilidad clara para los datos Ionosfera y Tictactoe, y moderada para Ripley; además, se percibe que para un número de neuronas pequeño (inferior a 4) se degradan las prestaciones del clasificador EST-MLP$_{\text{MLP}}$.

Estas cinco últimas figuras nos han proporcionado una información sobre la sensibilidad de nuestro diseño para completar este estudio, permitiendo concluir que el diseño EST-MLP$_{\text{MLP}}$ presenta una sensibilidad moderada con respecto a los parámetros a explorar por CV, a pesar de ser varios; siendo N_{ST} un parámetro delicado para un número pequeño de neuronas.

3.5. Conclusiones

En este capítulo, tras una breve revisión de las redes neuronales y especialmente del MLP, hemos presentado el diseño EST-MLP$_{\text{MLP}}$ basado en la idea de crear ESTs, realizando pruebas experimentales en tres problemas de clasificación bien conocidos (dos reales, Ionosfera y Tictactoe, y uno sintético, Ripley), comparando sus resultados con los de un MLP convencional y con otro diseñado empleando el método EDR de edición de muestras.

Los resultados llevan a concluir que, como se conjeturaba, el empleo de ESTs suficientemente generales y flexibles hace poco relevante la inclusión o exclusión de activaciones de salida en los MLPs entrenados con ellos, al tiempo que se obtienen prestaciones de clasificación superiores a las alternativas convencionales consideradas. También se ha verificado que la sensibilidad a los parámetros del EST es moderada.

Por lo anterior, queda abierto el camino para explorar el empleo del EST sobre otras familias de máquinas, y muy particularmente en el singular caso de los GPs; lo que se llevará a cabo en los capítulos que siguen.

Capítulo 4

Aplicación de los ESTs a Modelos de Mezclas de Gaussianas

4.1. Introducción

En este capítulo extendemos nuestra propuesta de ESTs sobre modelos generativos tipo GMMs para resolver problemas de clasificación. Esta aplicación es importante no solamente por la exploración de la aplicabilidad general de nuestra propuesta, sino también porque los GMMs son relativamente fáciles de interpretar y flexibles a la hora de ser aplicados.

Emplearemos GMMs de dos modos para diseñar una máquina de clasificación. El primero consiste en una modelización de la verosimilitud de una muestra $\mathbf{x}$ para cada clase C_i, $\widehat{p}(\mathbf{x}|C_i)$, como mezcla de un número apropiado de Gaussianas, y calcular las probabilidades *a priori* $\widehat{P}(C_i)$ como frecuencias relativas de cada clase; luego, se calcula la probabilidad *a posteriori*, $\widehat{P}(C_i|\mathbf{x})$, con el teorema de Bayes y, finalmente, se aplica un decisor MAP[1] para asignar una nueva muestra a su clase correspondiente. En el segundo, asumiendo que estamos trabajando con etiquetas blandas, se estima la distribución conjunta de la etiqueta $t_s(\mathbf{x})$ y de la

[1]En este caso particular del clasificador bayesiano, los parámetros de coste tienen la siguiente forma, $C_{00} = C_{11} = 1$ y $C_{01} = C_{10} = 0$.

muestra $\mathbf{x}$ como mezcla de un cierto número de Gaussianas. A continuación, se estima el valor de $t_s(\mathbf{x}^*)$, $\hat{t}_s(\mathbf{x}^*)$, mediante el estimador de mínimo MSE, que es la esperanza de la etiqueta continua (blanda) $t_s(\mathbf{x})$ dado el dato $\mathbf{x}^*$, $E\{t_s(\mathbf{x})|\mathbf{x}^*\}$. Finalmente, al presentar un nuevo dato, la clasificación se realiza mediante la aplicación de un decisor duro sobre la etiqueta blanda estimada.

El entrenamiento de los parámetros de un modelo GM se realiza con el algoritmo EM [Dempster1977] (véase Apéndice C). En lo que sigue, detallamos el diseño del clasificador basado en ESTs usando GMMs [El Jelali2008b], donde optamos por el modo discriminativo para estimar el valor de la etiqueta blanda $t_s(\mathbf{x})$.

4.2. Clasificación con modelos GMMs y ESTs

Construimos ambas máquinas; la primera, como referencia.

Para el primer diseño que llamaremos MAP GMM, se estima la verosimilitud de una muestra $\mathbf{x}$ del conjunto de entrenamiento,

$$\widehat{p}(\mathbf{x}|C_i) = \sum_{l=1}^{L} \pi_l \, p_l(\mathbf{x}|C_i), \qquad i = \pm 1 \tag{4.1}$$

para cada clase C_i como una mezcla de Gaussianas $\{p_l(\mathbf{x}|C_i)\}$; siendo L y π_l el número de Gaussianas y el factor de mezcla de la l-ésima componente $p_l(\mathbf{x}|C_i)$, respectivamente. Se optimizan los parámetros de la mezcla (el factor de la mezcla, el vector media y la matriz de covarianza correspondientes a la l-componente $p_l(\mathbf{x}|C_i)$) con el algoritmo EM. Al presentar una nueva muestra $\mathbf{x}^*$, se aplica el decisor MAP para obtener una decisión D_i:

$$\widehat{P}(C_1|\mathbf{x}^*) \underset{D_{-1}}{\overset{D_1}{\gtrless}} \widehat{P}(C_{-1}|\mathbf{x}^*)$$

donde

$$\widehat{P}(C_i|\mathbf{x}^*) = \frac{\widehat{p}(\mathbf{x}^*|C_i)\widehat{P}(C_i)}{\sum_{i'} \widehat{p}(\mathbf{x}^*|C_{i'})\widehat{P}(C_{i'})} \qquad i, i' = \pm 1. \tag{4.2}$$

Para diseñar la máquina basada en EST, la estimación de $t_s(\mathbf{x})$ se realiza bajo el modelo GMM multidimensional como sigue. Se asume que la distribución conjunta de $t_s(\mathbf{x})$ y $\mathbf{x}$ es una mezcla de L Gaussianas

$$\widehat{p}(t_s(\mathbf{x}), \mathbf{x}) = \sum_{l=1}^{L} \pi_l \, p_l(t_s(\mathbf{x}), \mathbf{x}) \tag{4.3}$$

donde $\{\pi_l\}$ $(0 \leq \pi_l \leq 1, \sum_l \pi_l = 1;\ l = 1, \ldots, L)$ y $\{p_l(t_s(\mathbf{x}), \mathbf{x})\}$ son los factores y las densidades de probabilidad de la mezcla, respectivamente; los parámetros de la mezcla se determinan mediante EM.

A continuación, la estimación de $t_s(\mathbf{x})$ se realiza mediante el estimador de mínimo MSE (obtenida como la esperanza *a posteriori* de $t_s(\mathbf{x})$ dada $\mathbf{x}$), obteniendo

$$\begin{aligned}
\widehat{t_s}(\mathbf{x}) &= \widehat{E}\{t_s(\mathbf{x})|\mathbf{x}\} = \int t_s(\mathbf{x}) \, \widehat{p}(t_s(\mathbf{x})|\mathbf{x}) \, dt_s \\
&= \int t_s(\mathbf{x}) \, \frac{\widehat{p}(t_s(\mathbf{x}), \mathbf{x})}{\widehat{p}(\mathbf{x})} \, dt_s = \sum_{l=1}^{L} \frac{\pi_l \, p_l(\mathbf{x})}{\sum_{l'=1}^{L} \pi_{l'} \, p_{l'}(\mathbf{x})} \int t_s(\mathbf{x}) \, \widehat{p}_l(t_s(\mathbf{x})|\mathbf{x}) \, dt_s \\
&= \sum_{l=1}^{L} \frac{\pi_l \, p_l(\mathbf{x})}{\sum_{l'=1}^{L} \pi_{l'} \, p_{l'}(\mathbf{x})} \, \widehat{E}_l\{t_s(\mathbf{x})|\mathbf{x}\}
\end{aligned} \tag{4.4}$$

donde las densidades marginales son

$$p_l(\mathbf{x}) = \int p_l(t_s(\mathbf{x}), \mathbf{x}) \, dt_s, \qquad \text{para} \ \ l = 1, \ldots, L. \tag{4.5}$$

Dado que las $\{p_l(t_s(\mathbf{x}), \mathbf{x})\}$ son Gaussianas, se puede verificar que la estimación óptima de MSE de $t_s(\mathbf{x})$ tiene forma lineal

$$\widehat{E}_l\{t_s(\mathbf{x})|\mathbf{x}\} = \mathbf{w}_{l,e}^{\mathrm{T}} \, \mathbf{x}_e = \mathbf{w}_l^{\mathrm{T}} \, \mathbf{x} + w_{0,l} \tag{4.6}$$

siendo $\mathbf{x}_e$ y $\mathbf{w}_{l,e}$ el vector de entrada y el vector de pesos extendidos, respectivamente; mientras que las $\{\mathbf{w}_l\}$ vienen dadas por las ecuaciones normales

$$\mathbf{w}_l = V_{\mathbf{xx},l}^{-1} \mathbf{v}_{t_s\mathbf{x},l}, \qquad l \neq 0 \tag{4.7}$$

siendo $V_{\mathbf{xx},l}$ y $\mathbf{v}_{t_s\mathbf{x},l}$ la matriz de auto-covarianza del dato $\mathbf{x}$, y el vector de la varianza cruzada de la etiqueta $t_s(\mathbf{x})$ y el dato $\mathbf{x}$ bajo el modelo Gaussiano $p_l(t_s(\mathbf{x}), \mathbf{x})$, respectivamente. $V_{\mathbf{xx},l}$ y $\mathbf{v}_{t_s\mathbf{x},l}$ se extraen a partir de la matriz de covarianza de esta distribución

$$V_{t_s\mathbf{x},l} = \begin{bmatrix} v_{t_s} & \mathbf{v}_{t_s\mathbf{x},l}^{\mathrm{T}} \\ \mathbf{v}_{t_s\mathbf{x},l} & V_{\mathbf{xx},l} \end{bmatrix} \tag{4.8}$$

donde v_{t_s} es la varianza de $t_s(\mathbf{x})$. Por otra parte, $w_{0,l}$ se calcula en la forma

$$w_{0,l} = E_l\{t_s(\mathbf{x})\} - \mathbf{w}_l^{\mathrm{T}}\, E_l\{\mathbf{x}\}. \tag{4.9}$$

$E_l\{t_s(\mathbf{x})\}$ y $E_l\{\mathbf{x}\}$ se extraen del vector media $\mathbf{m}_{t_s\mathbf{x},l}$ de la distribución de $p_l(t_s(\mathbf{x}), \mathbf{x})$ como

$$\mathbf{m}_{t_s\mathbf{x},l} = [m_{t_s,l} \quad \mathbf{m}_{\mathbf{x},l}]^{\mathrm{T}} \tag{4.10}$$

con $m_{t_s,l} = E_l\{t_s(\mathbf{x})\}$ y $\mathbf{m}_{\mathbf{x},l} = E_l\{\mathbf{x}\}$. Nótese que $E_l\{\mathbf{x}\}$ y $V_{\mathbf{xx},l}$ son el vector media y la matriz de covarianza de la densidad Gaussiana $p_l(\mathbf{x})$, respectivamente. La expresión (4.4) es similar a la correspondiente a la formulación de mezcla de Expertos (MoE, "Mixture of Experts") [Jacobs1991, Jiang1994], como previamente se mencionó en [Xu1995].

Finalmente, la decisión para una nueva muestra $\mathbf{x}^*$ es

$$dec(\mathbf{x}^*) = sgn(\widehat{t}_s(\mathbf{x}^*)) \tag{4.11}$$

La Figura 4.1 ilustra el diseño de nuestra máquina de decisión.

Denominaremos "EST-GMM$_{\mathrm{MLP}}$" al algoritmo de nuestro clasificador, que se describe del siguiente modo:

1. Entrenar una máquina auxiliar MLP$_{aux}$ con el conjunto de entrenamiento $\{\mathbf{x}^{(k)}, t(\mathbf{x}^{(k)})\}_{k=1}^{K}$.

2. Calcular la salida $o_{aux}(\mathbf{x}^{(k)})$ y el error $e(\mathbf{x}^{(k)})$ de la máquina auxiliar previamente entrenada para la k-ésima muestra.

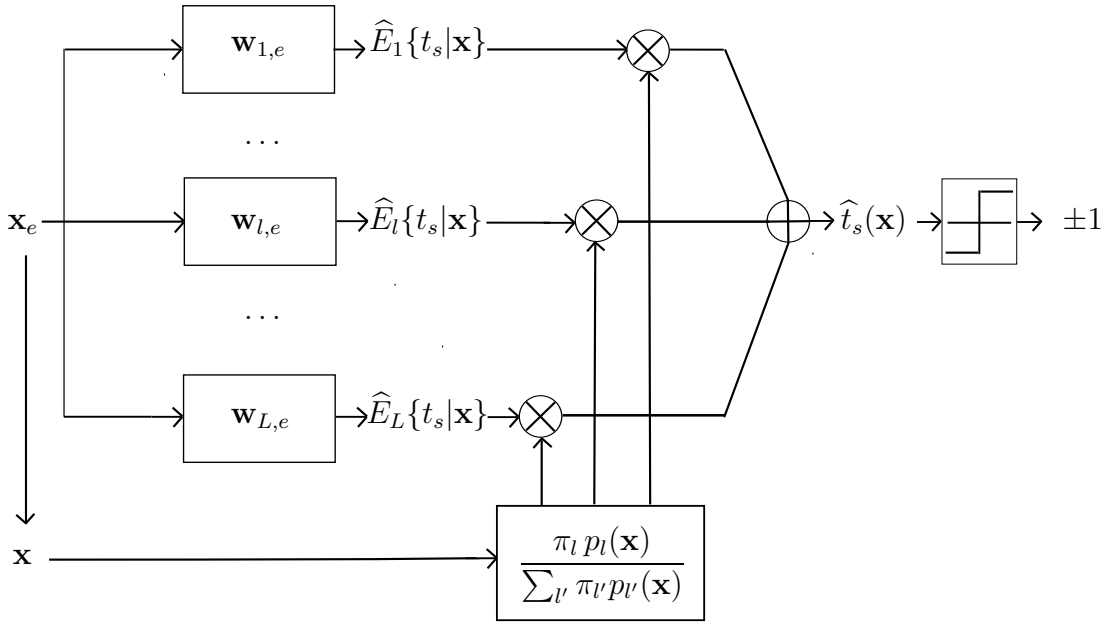

Figura 4.1: Esquema del diseño del clasificador usando ESTs y modelos GMMs.

3. Determinar el conjunto $\{\mathbf{x}^{(k)}, t_s(\mathbf{x}^{(k)})\}_{k=1}^{K}$ como

$$t_s(\mathbf{x}^{(k)}) \;=\; \lambda(|e(\mathbf{x}^{(k)})|)\; t(\mathbf{x}^{(k)}) \;+\; (\,1 - \lambda(|e(\mathbf{x}^{(k)})|)\,)\; \mathrm{o}_{aux}(\mathbf{x}^{(k)}) \qquad (4.12)$$

con

$$\lambda(|e(\mathbf{x}^{(k)})|) = \begin{cases} \exp(-\dfrac{(|e(\mathbf{x}^{(k)})| - \mu)^2}{\alpha_1}) & \text{para} \quad |e(\mathbf{x}^{(k)})| \le \mu \\ \\ \exp(-\dfrac{(|e(\mathbf{x}^{(k)})| - \mu)^2}{\alpha_2}) & \text{para} \quad \mu < |e(\mathbf{x}^{(k)})| \le 2 \end{cases} \qquad (4.13)$$

siendo μ, α_1 y α_2 los parámetros libres de las campanas de Gauss, que se determinan por CV.

4. Optimizar los parámetros de la distribución $\widehat{p}(t_s(\mathbf{x}^{(k)}), \mathbf{x}^{(k)})$, $\{\pi_l,\ \mathbf{m}_{t_s\mathbf{x}^{(k)},l},\ V_{t_s\mathbf{x}^{(k)},l}\}_{l=1}^{L}$ mediante el algoritmo EM.

5. Calcular $\widehat{t}_s(\mathbf{x}^{(k)})$ con el estimador de mínimo MSE como sigue:

   a) Extraer de $V_{t_s\mathbf{x},l}$, la matriz $V_{\mathbf{xx},l}$ y el vector $\mathbf{v}_{t_s\mathbf{x},l}$

b) Calcular $\mathbf{w}_l$ con la fórmula (4.7)

c) Extraer de $\mathbf{m}_{t_s\mathbf{x},l}$, la esperanza de $t_s(\mathbf{x})$, $E_l\{t_s(\mathbf{x})\}$, y el vector $E_l\{\mathbf{x}\}$

d) Calcular $w_{0,l}$ con la ecuación (4.9)

e) Estimar $\widehat{E}_l\{t_s(\mathbf{x}^{(k)})|\mathbf{x}^{(k)}\}$ con (4.6) con $\mathbf{x}_e = \mathbf{x}_e^{(k)}$

f) Determinar las marginales $p_l(\mathbf{x})$ con la fórmula (4.5)

g) Estimar $t_s(\mathbf{x}^{(k)})$ con la expresión (4.4) con $\mathbf{x} = \mathbf{x}^{(k)}$

4.3. Pruebas experimentales

4.3.1. Conjuntos de datos

Hemos trabajado con seis bases de datos utilizadas frecuentemente en clasificación. Kwok (**kwo**) [Kwok1999] es un problema sintético bidimensional. La tasa de error del clasificador bayesiano es 11.3 %. Los otros cinco problemas son datos reales obtenidos del repositorio de UCI [Blake]: Abalone (convertido a un problema binario de acuerdo con [Ruiz2001]), Breast Cancer, Contraceptive, Ionosfera y Pima Indian. Nos referiremos a estos problemas como **aba**, **bre**, **con**, **ion**, y **pim**, respectivamente. La Tabla 4.1 describe sus principales características.

4.3.2. Entrenamiento y resultados

Todos los datos están normalizados entre -1 y 1. Utilizamos un MLP con N neuronas en la capa oculta como máquina auxiliar, y también entrenamos un MLP convencional de N' neuronas ocultas, como sistema de referencia en la tabla de resultados (el MLP auxiliar y el MLP convencional se entrenan de la misma manera que la máquina auxiliar del diseño EST-MLP$_{\text{MLP}}$ y el MLP convencional empleados en el Capítulo 3, respectivamente). Tanto N y N' como los parámetros μ, α_1 y α_2, el número de componentes de GMM, L para el modelo conjunto y L_1 y L_{-1} para los modelos de cada clase separada, se determinan por CV de 10 particiones (90 % entrenamiento y 10 % validación), con 10 realizaciones, explorando los siguientes valores:

Problemas	Entrenamiento (C_{+1}/C_{-1})	Test (C_{+1}/C_{-1})	Dimensión
aba	2507 (1238/1269)	1670 (843/827)	8
bre	420 (145/275)	279 (96/183)	9
con	883 (506/377)	590 (338/252)	9
ion	201 (101/100)	150 (124/26)	34
kwo	500 (200/300)	10200 (4080/6120)	2
pim	461 (161/300)	307 (107/200)	8

Tabla 4.1: Principales características de los problemas de clasificación utilizados en la parte experimental.

- N, N': 4, 6, 8, 10 ,12, 14, 16.
- μ: 0.01, 0.1, 0.3, 0.6, 1, 1.2, 1.6, 2.
- α_1, α_2: 0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 3, 4, 5.
- L: 4, 5, 6, 7, 8, 9, 10.
- L_1, L_{-1}: 2, 3, 4, 5.

Durante los experimentos no nos hemos enfrentado al problema de singularidad de la matriz de covarianza [Archambeau2003]; sin embargo, si este problema surgiese, podríamos remediarlo utilizando la solución propuesta en [Archambeau2004].

También, proporcionamos los resultados de la SVM de núcleo Gaussiano cuyos parámetros, el factor de penalización C y la dispersión σ, se determinan por CV con 10 particiones, explorando los siguientes valores:

- C: 10^{-1}, 1, 10, 10^2, 10^3, 10^4.

- σ: $\sqrt{D} \times \{2^{-5},\ 2^{-4},\ 2^{-3},\ 2^{-2},\ 2^{-1},\ 1,\ 2,\ 2^2,\ 2^3,\ 2^4,\ 2^5\}$, siendo D la dimensión del dato de entrada.

La implementación utilizada para simular las SVMs es la toolbox Matlab IRWLS SVM para reconocimiento de patrones [Pérez-Cruz2001] disponible en www.tsc.uc3m.es/∼fernando/, con tolerancia de 10^{-5} para proporcionar una buena precisión a la solución del algoritmo de programación cuadrática.

La Tabla 4.2 presenta la tasa de acierto de clasificación del MLP convencional, del MAP GMM, del EST-GMM$_{\text{MLP}}$, y de la máquina SVM con los valores de los parámetros de diseño correspondientes encontrados por CV, y también los resultados de la aproximación "omnisciente", que corresponde al diseño máquina que ofrece la mejor tasa de acierto de clasificación sobre datos de test. Está claro, que este procedimiento no nos permite considerar estos diseños como válidos; sin embargo, esta aproximación sirve para medir la capacidad máxima de los diseños.

Con respecto al MLP convencional, los resultados de EST-GMM$_{\text{MLP}}$ muestran ventaja para los problemas **ion** y **kwo**, y hay casi igualdad para **pim**. De la comparación entre EST-GMM$_{\text{MLP}}$ y MAP GMM, resulta que EST-GMM$_{\text{MLP}}$ presenta una ligera ventaja para **aba**, **bre**, y **kwo**, y muestra una ventaja importante para **con**, **ion**, y **pim** con respecto a MAP GMM. También observamos que los resultados de EST-GMM$_{\text{MLP}}$ son competitivos con los de la SVM, habiendo ventaja para los problemas **aba**, **kwo**, y **pim**. Los resultados llevan a concluir que la máquina EST-GMM$_{\text{MLP}}$ es competitiva con respecto a los diseños estándares del MLP y de la SVM, y supera en todos los problemas considerados los resultados correspondientes a MAP GMM.

La Figura 4.2 presenta las fronteras de clasificación sobre los datos de test del problema **kwo**, obtenidas con los siguientes cuatro clasificadores: el EST-GMM$_{\text{MLP}}$, el MAP GMM, el MLP convencional, y finalmente, el clasificador de Bayes (la frontera teórica). Los datos de test de kwok tienen la siguiente forma: una mezcla de dos Gaussianas para la clase $+1$ (círculos) y una mezcla de tres Gaussianas para la clase -1 (puntos); por claridad, los datos de test se han submuestreado a 500 muestras (C_{+1}:200/C_{-1}:300). En dicha figura, se percibe que la frontera de EST-GMM$_{\text{MLP}}$ se ajusta más a la frontera teórica que las fronteras

Problemas	**aba**	**bre**	**con**	**ion**	**kwo**	**pim**
MLP CV	**78.12±0.54**	97.51±0.60	70.38± 2.24	93.22±1.48	83.49±3.32	**78.33±1.33**
N'	12	8	10	6	16	6
MLP omni*	78.12±0.54	97.51±0.60	70.38± 2.24	93.22±1.48	83.49±3.32	78.33±1.33
N'	12	8	10	6	16	6
MAP GMM CV	72.90±0.84	94.90±1.46	62.88±1.41	93.15±2.56	84.77±0.59	72.72±1.40
L_1/L_{-1}	4/4	3/4	4/4	3/3	3/3	2/2
MAP GMM omni*	72.97±0.58	95.94±1.08	63.15±0.95	93.15±2.56	85.30±0.62	72.89±1.80
L_1/L_{-1}	5/5	5/4	2/2	3/3	2/5	2/3
EST-GMM$_{\text{MLP}}$ CV	73.01±0.67	95.34±0.91	65.69±1.98	94.52±1.82	**85.00±0.83**	77.37±1.50
$L/N/\mu/$	9/12/1.2/	8/10/0.3/	8/6/1.6/	9/6/1.2/	9/6/0.3/	6/4/0.3/
α_1/α_2	0.1/0.1	0.01/0.05	0.1/4	0.05/0.5	0.01/0.1	0.01/1
EST-GMM$_{\text{MLP}}$ omni*	73.01±0.67	95.77±1.10	65.69±1.98	94.54±1.89	85.31±0.62	78.92±0.98
$L/N/\mu/$	9/12/1.2/	6/10/0.3/	8/6/1.6/	10/8/1/	6/12/0.3/	6/12/0.3/
α_1/α_2	0.1/0.1	10^{-3}/1.5	0.1/4	5/2	0.1/0.01	0.1/0.01
SVM CV	66.73±3.92	**97.31±0.25**	**70.76±0.38**	**97.80±0.45**	84.43±0.67	72.35±1.24
C/σ	$10^4/2^{-1}\sqrt{D}$	$1/\sqrt{D}$	$10^3/2\sqrt{D}$	$10/2^{-1}\sqrt{D}$	$10/2^{-3}\sqrt{D}$	$10^2/2^{-1}\sqrt{D}$
SVM omni*	77.84±2.44	97.92±0.23	71.29±0.72	97.80±0.45	85.26±0.75	79.77±0.95
C/σ	$10^4/2^5\sqrt{D}$	$10/\sqrt{D}$	$100/\sqrt{D}$	$10/2^{-1}\sqrt{D}$	$10^4/2^{-1}\sqrt{D}$	$10^2/2^2\sqrt{D}$

Tabla 4.2: Tasa de acierto de clasificación ($\pm$ desviación estándar) sobre datos de test de **aba**, **bre**, **con**, **ion**, **kwo**, and **pim**, y parámetros de diseño de cada método (MLP: N'; MAP GMM: L_1, L_{-1}; EST-GMM$_{\text{MLP}}$: $L, N, \mu, \alpha_1, \alpha_2$; y SVM: C) seleccionados por CV. "omni" se refiere a los resultados de la aproximación "omnisciente" sobre estas máquinas.

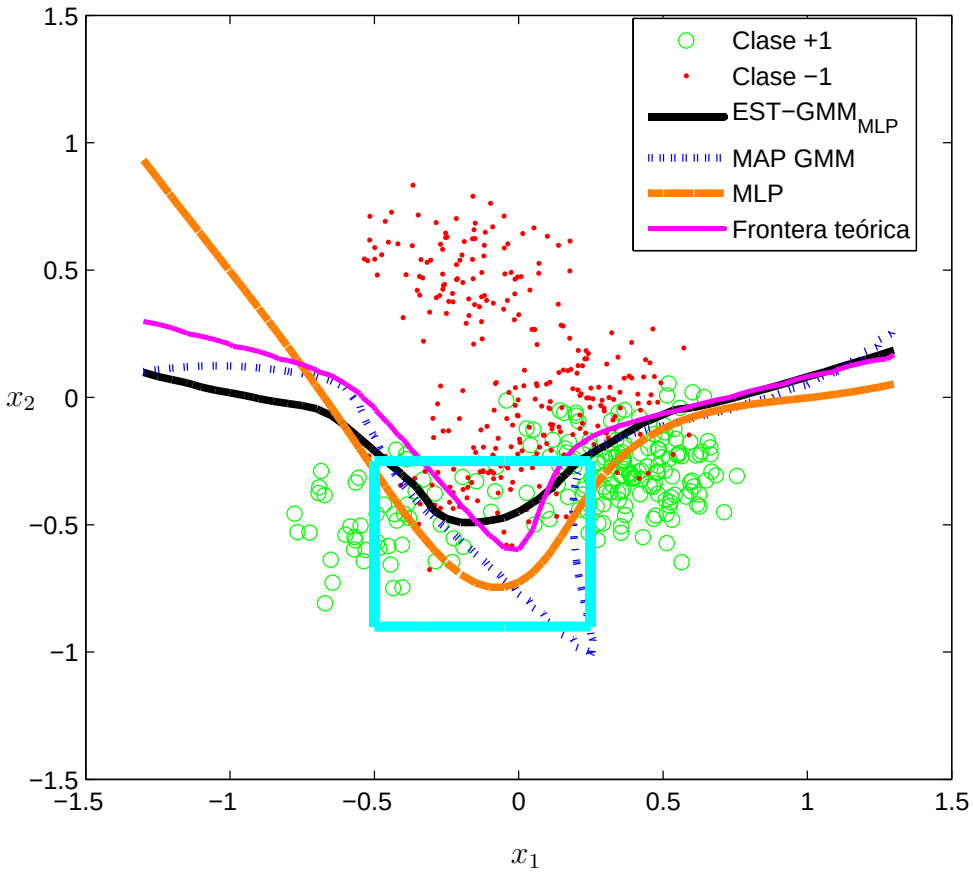

Figura 4.2: Fronteras de los tres métodos mencionados en la Tabla 4.2 comparados con la frontera teórica del problema kwo con datos de test sub-muestreados a 500 muestras (C_{+1} : 200/C_{-1} : 300).

correspondientes a MAP GMM y a MLP: la máquina EST-GMM$_{\text{MLP}}$ clasifica bien algunas muestras a las que los clasificadores MAP GMM y MLP consideran en la región de decisión errónea, como se ve en la zona definida por $-0.5 \leq x_1 \leq 0.25$.

Está claro, también, que la ventaja obtenida con nuestra propuesta conlleva un coste adicional durante el proceso de entrenamiento, en cual necesitamos seleccionar los mejores valores de diseño: $\#N \times \#\mu \times \#\alpha_1 \times \#\alpha_2 \times \#L$. Por otro lado, dicho coste es reducido para un MLP estándar ($\#N'$), MAP GMM ($\#L_1 \times \#L_{-1}$) y SVM ($\#C \times \#\sigma$). Desde luego, el elevado coste computacional

se limita solamente a la etapa de entrenamiento: una vez entrenada la máquina, su aplicación requiere un esfuerzo computacional equivalente al caso de GMM para regresión.

Otra dificultad adicional de nuestra propuesta que puede aparecer en determinados problemas es la sensibilidad con respecto a los valores de los parámetros de diseño seleccionados por CV. Consideramos los diseños "omniscientes" para averiguar la sensibilidad del diseño con respecto a los parámetros obtenidos por CV.

Se observa en la Tabla 4.2 que, incluso teniendo que seleccionar muchos más parámetros, el diseño óptimo por CV y el omnisciente obtenidos coinciden para los problemas **aba** y **con** en EST-GMM$_{\text{MLP}}$ y para MAP GMM en **ion**. Además, aunque los parámetros de diseño obtenidos por las dos vías (CV y "omnisciente") son diferentes, las prestaciones del EST-GMM$_{\text{MLP}}$ correspondientes a la CV y del "omnisciente" presentan una ligera diferencia en los problemas **bre**, **ion**, y **kwo** y una diferencia significativa sólo para **pim**. Hay que resaltar que, para este último problema, precisamente, nuestro diseño obtiene claras mejoras con respecto a los diseños estándares. Cuando aplicamos los esquemas MAP GMM, también hay una pequeña diferencia para **kwo** y **con**, y es importante para **bre**. Para los MLPs, CV ofrece el resultado de la aproximación omnisciente en **ion**, y hay una diferencia significativa para **kwo** y **pim**, pese a que, para el MLP, N' es el único parámetro a explorar.

Finalmente, para completar esta análisis sobre la sensibilidad, las Figuras 4.3, 4.4, 4.5, 4.6, y 4.7 ilustran la sensibilidad con respecto a los parámetros L, N, μ, α_1, y α_2 alrededor del punto que corresponde al diseño CV (representado por los cuadros) sobre datos de test de los problemas de la Tabla 4.1.

Con respecto al parámetro L, las curvas de sensibilidad de **bre**, **con**, y **pim** son casi planas, y se nota una ligera sensibilidad para los problemas **ion** y **kwo**, mientras la curva correspondiente a **aba** presenta una sensibilidad notable, aunque el punto correspondiente al diseño por CV se encuentra en una zona de altas prestaciones, y esto se ve para todos los problemas.

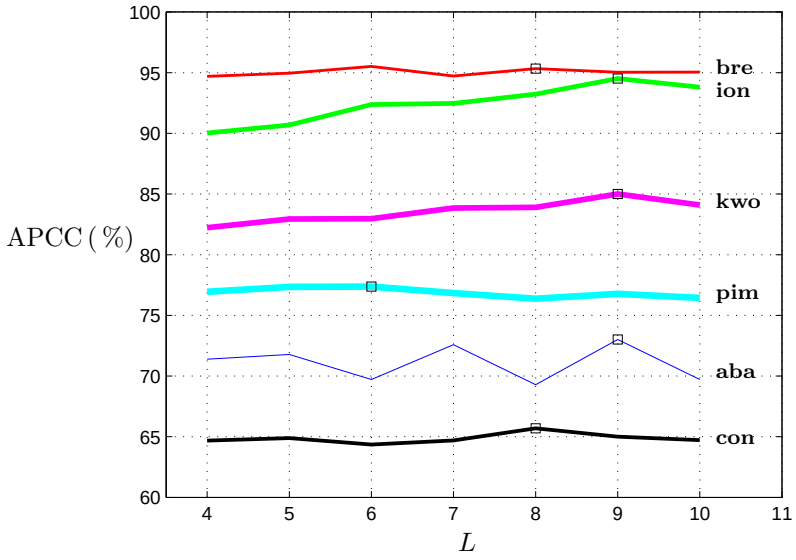

Figura 4.3: Sensibilidad respecto a L de EST-GMM$_{\text{MLP}}$ sobre los datos de test de los problemas de la Tabla 4.1. APCC ("Average Percentage of Correct Classification") es la tasa de acierto de clasificación en %. Los cuadros representan los valores encontrados por CV.

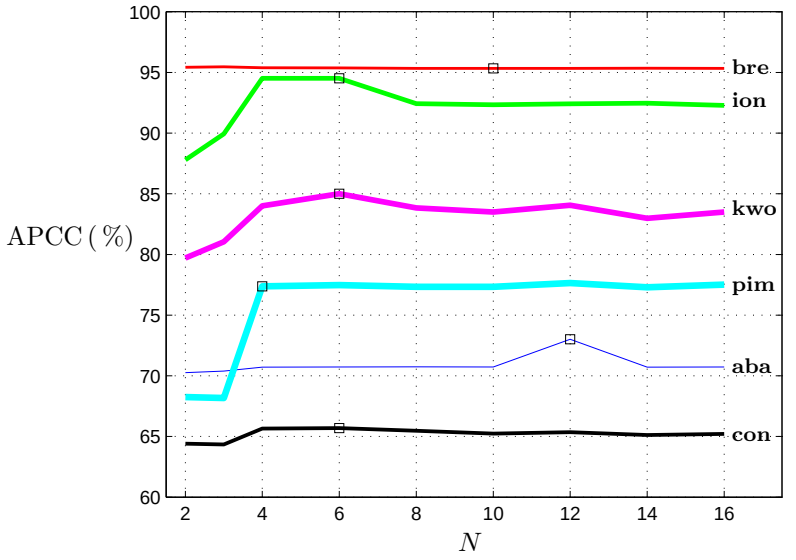

Figura 4.4: Sensibilidad respecto a N de EST-GMM$_{\text{MLP}}$ sobre los datos de test de los problemas de la Tabla 4.1.

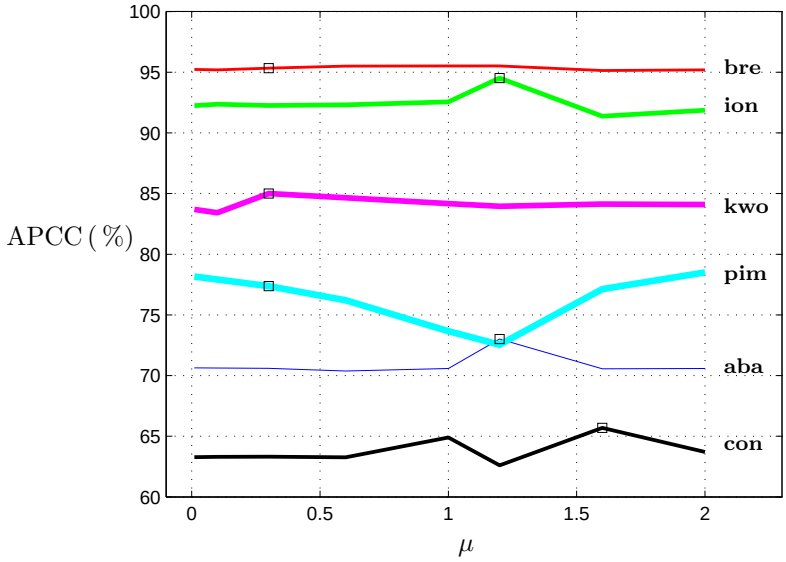

Figura 4.5: Sensibilidad respecto a μ de EST-GMM$_{\text{MLP}}$ sobre los datos de test de los problemas de la Tabla 4.1. (El cuadro donde se cruzan las curvas de sensibilidad de los problemas **pim** y **aba**, pertenece a la curva de sensibilidad del **aba**).

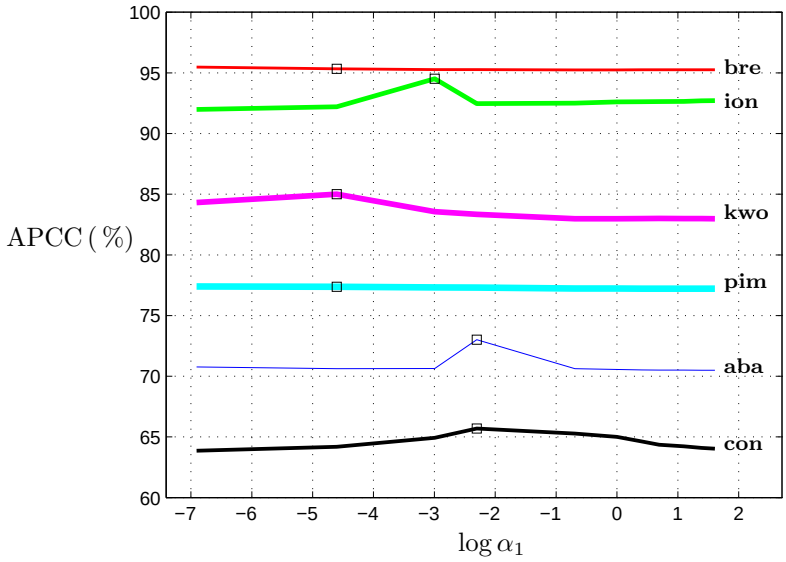

Figura 4.6: Sensibilidad respecto a α_1 de EST-GMM$_{\text{MLP}}$ sobre los datos de test de los problemas de la Tabla 4.1.

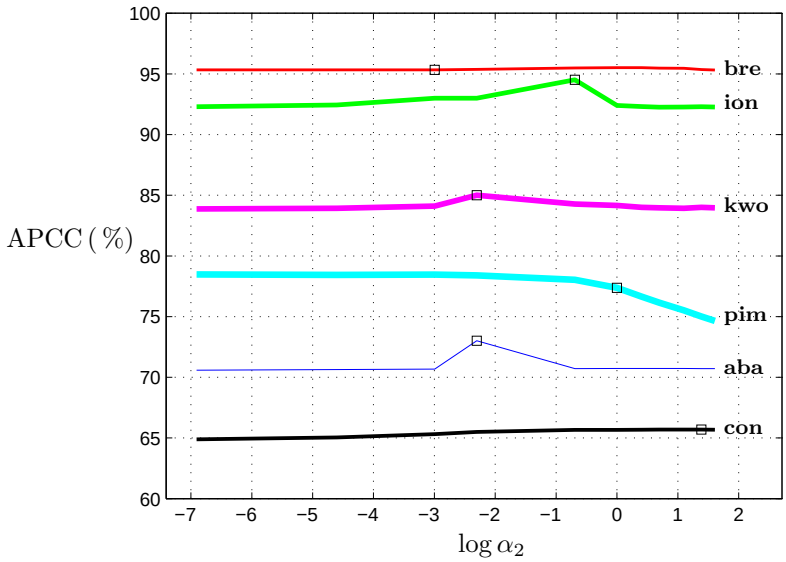

Figura 4.7: Sensibilidad respecto a α_2 de EST-GMM$_{\text{MLP}}$ sobre los datos de test de los problemas de la Tabla 4.1.

Con respecto a N, hay que resaltar que para valores inferiores a 4 las prestaciones de la máquina decrecen para los siguientes problemas: **con**, **ion**, **kwo**, y **pim**. La curva correspondiente al caso **bre** es plana, y hay una sensibilidad moderada para **aba**, **con**, **ion**, **kwo**, y **pim**; el punto correspondiente al diseño CV está siempre en la región de altas prestaciones.

A partir de la Fig. 4.5, se nota que μ es un parámetro delicado en cuanto a sensibilidad, y eso se observa en las curvas de **con**, **ion**, y **pim**: para valores de $1 \leq \mu \leq 1.5$, hay variaciones notables en términos de prestaciones. Para **aba**, **bre**, y **kwo**, hay una ligera sensibilidad respecto a μ.

Finalmente, con respecto a los parámetros α_1 y α_2, se nota para casi todos los problemas (**aba**, **bre**, **con**, **ion**, y **kwo**) que existe una baja sensibilidad; sin embargo, para **pim**, las prestaciones decrecen para valores grandes de α_1.

Como recapitulación en cuanto a la discusión sobre la sensibilidad, podemos decir que el diseño EST-GMM$_{\text{MLP}}$ presenta una ligera sensibilidad con respecto a L, α_1, y α_2 , y con respecto a N, cabe destacar que las prestaciones se degradan para números pequeños de neuronas ocultas. μ es también un parámetro delicado

para valores entre 1 y 1.5. Finalmente, podemos concluir que nuestro método basado en la idea EST presenta una sensibilidad moderada aunque en CV el número de parámetros a explorar es alto.

4.4. Conclusiones

En este capítulo hemos aplicado la idea de énfasis a los modelos GMMs para tareas de decisión (también es posible extender la idea a otros modelos generativos, como son las ventanas de Parzen [Nadaraya1964], usando la regresión de Nadaraya-Watson [Parzen1962, Watson1990] con la etiqueta EST). Hemos usado formulaciones básicas para diseñar las máquinas MAP GMM y EST-GMM$_{\text{MLP}}$. Los resultados de la parte experimental demuestran que el diseño EST-GMM$_{\text{MLP}}$ proporciona regularmente ventaja con respecto a la máquina MAP GMM para todos los problemas considerados en la parte experimental, y es competitivo con respecto a clasificadores convencionales como los MLPs y las SVMs. Se comprueba así que la idea basada en el mecanismo atencional EST aplicado a los GMMs es eficaz para resolver tareas de clasificación, ofreciendo una baja sensibilidad respecto a los parámetros a fijar por CV pese al elevado número de éstos. Obviamente, se necesita un esfuerzo computacional adicional para el diseño del clasificador basado en EST; pero, una vez entrenada la máquina, su aplicación es computacionalmente ligera, y, como los GMMs son generativos, ofrece las ventajas de que permiten una interpretación fácil y la posibilidad de aplicar los principales métodos para trabajar con datos imputados.

Capítulo 5

Diseño de clasificadores tipo GP mediante las técnicas EST

5.1. Introducción

Los GPs -una buena exposición general es [Rasmussen2006]- son máquinas de aprendizaje que, en su versión para regresión (máquinas GPR, "Gaussian Process Regression"), constituyen una extensión inmediata del filtro (discreto) de Wiener [Wiener1948], trabajando sobre espacios multidimensionales (de las observaciones $\mathbf{x}$) y con muestras irregularmente distribuidas, cuya dificultad solventan recurriendo al "truco del núcleo", proponiendo una forma parametrizada para las funciones de covarianza. Su creciente popularidad y su abundante uso se deben a la robustez que se deriva del hecho de emplear un número reducido de parámetros para dichas covarianzas; como además esos parámetros se aprenden a partir de una formulación de máxima verosimilitud, queda prácticamente descartada la posibilidad de aparición de sobreajuste.

Ventaja adicional de las máquinas GPR es que su formulación proporciona, como veremos, un indicador de la fiabilidad de sus resultados mediante la varianza de la variable que en el modelo representa los valores a estimar. Desafortunadamente, la inversión matricial característica de la aproximación de Wiener, que en

el caso de los GPs no implica una matriz Toeplitz por no asumirse estacionaridad y por la irregularidad de las muestras, conlleva una carga computacional de modelado muy alta, ya que necesita $O(K^3)$ operaciones; mientras que la estimación de cada valor predicho requiere $O(K^2)$.

Por su propia naturaleza, los GPs no pueden ser empleados directamente para resolver problemas de clasificación; lo cual ha llevado a la utilización de variables latentes y la posterior aplicación de aproximaciones (y métodos iterativos, junto con numéricos), como son los de Laplace [Williams1998] y los métodos EP ("Expectation-Propagation") [Minka2001a] y EM-EP ("Expectation Maximization-Expectation Propagation") [Kim2006], para construir máquinas GPC ("Gaussian Process Classification"). Sus resultados, en todo caso, son de calidad moderada. Otra alternativa, computacionalmente muy costosa, es recurrir a métodos MonteCarlo, como el MonteCarlo de Cadenas de Markov [Neal1993].

En tal contexto, la aplicación de los métodos EST resulta natural, ya que sustituir por blancos blandos enfatizados las etiquetas duras permite aplicar la sencilla formulación GPR y mantener intactas todas sus ventajas, al tiempo que un adecuado énfasis posibilita obtener resultados competitivos. A ello se dedicará este Capítulo, que tiene relevancia central en esta Tesis, precisamente por las dificultades con que se encuentra el desarrollo de máquinas GPC por las vías seguidas hasta ahora.

Este Capítulo se organiza del siguiente modo. En primer lugar, aparece un conciso resumen de la formulación GPR y se presenta la máquina EST-GP. Después, se revisan brevemente las máquinas GPC y las tres aproximaciones mencionadas anteriormente. A continuación, se describen el diseño de las máquinas empleadas en la parte experimental y los experimentos, y se hace una discusión detallada de sus resultados (incluyendo la carga computacional de los diseños considerados y un estudio de la sensibilidad de los EST-GPs con respecto a sus parámetros libres). Finalmente, se cierra el Capítulo con una síntesis de las conclusiones extraídas de la parte experimental.

5.2. Aplicación de ESTs para el diseño de clasificadores GP

Un GP real $f(\mathbf{x})$ queda definido por su función media

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{5.1}$$

y su función covarianza

$$c(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{5.2}$$

siendo $\mathbf{x}$ y $\mathbf{x}'$ dos vectores de entrada de dimensión $D \times 1$. Por razón de sencillez, la función media $f(\mathbf{x})$ se considera nula, una vez detraída la media muestral de los valores de las muestras.

En el caso de las máquinas GPR, las etiquetas $t(\mathbf{x})$ se asumen continuas y se modelan como un GP de media nula y de covarianza $c_{\boldsymbol{\theta},\theta_n}(\mathbf{x}, \mathbf{x}') = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') + \theta_n \delta_{\mathbf{x},\mathbf{x}'}$; siendo $k_{\boldsymbol{\theta}}$ y $\delta_{\mathbf{x},\mathbf{x}'}$ la función núcleo con vector $\boldsymbol{\theta}$ de parámetros a fijar y la función delta de Kronecker[1], respectivamente. Los valores de $\boldsymbol{\theta}$ y θ_n (θ_n es el nivel del ruido) se determinan durante el entrenamiento de la forma que veremos más adelante.

Obviamente, la distribución conjunta del vector de etiquetas de las muestras disponibles $\mathbf{t} = \{t(\mathbf{x}^{(k)})\}_{k=1}^{K}$ y la predicción $t^* = t(\mathbf{x}^*)$ correspondiente a una nueva observación $\mathbf{x}^*$ es también gaussiana con vector media nulo y matriz de covarianza

$$\mathbf{C}_{\boldsymbol{\theta},\theta_n \mathrm{C}} = \begin{bmatrix} \mathbf{C}_{\boldsymbol{\theta},\theta_n} & \mathbf{k}^*_{\boldsymbol{\theta}} \\ \mathbf{k}^{*\mathrm{T}}_{\boldsymbol{\theta}} & c^{**}_{\boldsymbol{\theta},\theta_n} \end{bmatrix} \tag{5.3}$$

donde $\mathbf{C}_{\boldsymbol{\theta},\theta_n}$ es una matriz de dimensión $K \times K$ y de elementos $k_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \delta_{\mathbf{x}^{(i)},\mathbf{x}^{(j)}}$, $\mathbf{k}^*_{\boldsymbol{\theta}}$ es un vector con componentes $k_{\boldsymbol{\theta}}(\mathbf{x}^*, \mathbf{x}^{(j)})$, y $c^{**}_{\boldsymbol{\theta},\theta_n}$ es la autocovarianza $k_{\boldsymbol{\theta}}(\mathbf{x}^*, \mathbf{x}^*) + \theta_n$.

Mediante una simple operación matricial (véase el Apéndice D) se deduce la forma de la distribución predictiva de t^* condicionada al conjunto de entrenamiento $\mathcal{D}$ y a la nueva observación $\mathbf{x}^*$, que es una gaussiana:

$$p(t^*|\mathbf{x}^*, \mathcal{D}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\mathbf{k}^{*\mathrm{T}}_{\boldsymbol{\theta}}\ \mathbf{C}^{-1}_{\boldsymbol{\theta},\theta_n}\mathbf{t}, c^{**}_{\boldsymbol{\theta},\theta_n} - \mathbf{k}^{*\mathrm{T}}_{\boldsymbol{\theta}}\ \mathbf{C}^{-1}_{\boldsymbol{\theta},\theta_n}\ \mathbf{k}^*_{\boldsymbol{\theta}}) \tag{5.4a}$$

[1] $\delta_{\mathbf{x},\mathbf{x}'} = 1$ si $\mathbf{x} = \mathbf{x}'$, y 0 en cualquier otro caso.

siendo

$$\hat{t}^* = E[t^*|\mathcal{D}] = \mathbf{k}_{\boldsymbol{\theta}}^{*\mathrm{T}}\ \mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\mathbf{t} \tag{5.4b}$$

y

$$var[t^*|\mathcal{D}] = c_{\boldsymbol{\theta},\theta_n}^{**} - \mathbf{k}_{\boldsymbol{\theta}}^{*\mathrm{T}}\ \mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\ \mathbf{k}_{\boldsymbol{\theta}}^{*} \tag{5.4c}$$

la estimación MSE de t^* y su varianza, respectivamente; dicha varianza proporciona una indicación de la confidencia del estimador. Esto es una de las ventajas de los GPs; también destaca la resistencia al sobreajuste si la función núcleo seleccionada tiene un número reducido de hiperparámetros.

El aprendizaje en los GPs consiste en determinar la forma adecuada de la función de covarianza $c_{\boldsymbol{\theta},\theta_n}$ (véase la parte experimental) y los valores del vector $\boldsymbol{\theta}$ y θ_n. Los valores de $\boldsymbol{\theta}$ y θ_n se determinan maximizando el logaritmo de la verosimilitud marginal $\log p(\mathbf{t}|\mathbf{X},\boldsymbol{\theta},\theta_n)$ con respecto a $\boldsymbol{\theta}$ y θ_n ($\mathbf{X} = \{\mathbf{x}^{(k)}\}_{k=1}^{K}$). Como $p(\mathbf{t}|\mathbf{X},\boldsymbol{\theta},\theta_n)$ es una gaussiana de vector media nulo y matriz de covarianza $\mathbf{C}_{\boldsymbol{\theta},\theta_n}$, la forma analítica de $\log p(\mathbf{t}|\mathbf{X},\boldsymbol{\theta},\theta_n)$ es

$$\log p(\mathbf{t}|\mathbf{X},\boldsymbol{\theta},\theta_n) = -\frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\mathbf{t} - \frac{1}{2}\log|\mathbf{C}_{\boldsymbol{\theta},\theta_n}| - \frac{K}{2}\log 2\pi \tag{5.5}$$

y las derivadas parciales de la expresión (5.5) con respecto a θ (uno de los hiperparámetros) son

$$\frac{\partial \log p(\mathbf{t}|\mathbf{X},\boldsymbol{\theta},\theta_n)}{\partial\theta} = \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta}\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\mathbf{t} - \frac{1}{2}tr(\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta}) \tag{5.6}$$

Es fácil probar que $\dfrac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}}{\partial\theta} = -\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\dfrac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta}\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}$ y $\dfrac{\partial \log|\mathbf{C}_{\boldsymbol{\theta},\theta_n}|}{\partial\theta} = tr(\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}\dfrac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta})$.

El cálculo del primer término de la ecuación (5.5) necesita invertir la matriz $\mathbf{C}_{\boldsymbol{\theta},\theta_n}$, de dimensión $K\times K$; según los métodos estándares, dicha inversión requiere un esfuerzo del orden de $O(K^3)$. Dada $\mathbf{C}_{\boldsymbol{\theta},\theta_n}^{-1}$, el cálculo de las derivadas en (5.6) requiere solamente un tiempo del orden de $O(K^2)$ para cada hiperparámetro. A veces, la maximización presenta problemas por la existencia de múltiples óptimos locales.

Para diseñar una máquina EST-GP para clasificación, la idea es sencilla: aplicar el procedimiento anterior directamente con etiquetas blandas.

Por otra parte, en el caso de las máquinas GPC, las etiquetas son discretas; por consiguiente, no podemos aplicar las formulaciones de regresión. Por lo tanto, es necesario introducir una variable "latente" continua $f(\mathbf{x})$. Para el caso binario, se aplica la función logística de regresión para obtener la probabilidad de observar la etiqueta $t(\mathbf{x}) = 1$ dada $f(\mathbf{x})$ como

$$P(t(\mathbf{x}) = 1|f(\mathbf{x})) = \frac{1}{1 + \exp(-f(\mathbf{x}))} = sgm(f(\mathbf{x})) \tag{5.7}$$

siendo $sgm(.)$ la función sigmoide, $sgm(.) = 1/(1 + \exp(.))$.

La variable latente $f(\mathbf{x})$ se asume gaussiana de vector media nulo y de matriz de covarianza $\mathbf{C}_{\boldsymbol{\theta},\theta_n}$, y se aplica sobre ella el procedimiento propio de la regresión.

Para calcular $P(t^* = 1|f^*)$, la inferencia se realiza en dos etapas.

En primer lugar, se calcula la distribución de la variable latente f^* para una muestra $\mathbf{x}^*$:

$$p(f^*|\mathbf{x}^*, \mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \int p(f^*|\mathbf{x}^*, \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)\, p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)\, d\mathbf{f} \tag{5.8}$$

siendo $\mathbf{f} = [f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(K)})]^T$.

En segundo lugar, se utiliza la distribución de f^* para obtener $P(t^* = 1|f^*)$:

$$P(t^* = 1|f^*) = \int sgm(f^*)\, p(f^*|\mathbf{x}^*, \mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)\, df^* \tag{5.9}$$

La aplicación de la ecuación (5.8) depende del cálculo de los términos $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ y $p(f^*|\mathbf{x}^*, \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$. El primero se obtiene según la regla de Bayes:

$$p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) \propto p(\mathbf{t}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) = \prod_{k=1}^{K} sgm(f(\mathbf{x}^{(k)}))\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) \tag{5.10}$$

Afortunadamente, el cálculo de $p(f^*|\mathbf{x}^*, \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ es sencillo, y corresponde a condicionar respecto a $\mathbf{f}$ la distribución gaussiana conjunta de $\mathbf{f}$ y f^* dadas las

observaciones $\mathbf{X}$ y la nueva muestra $\mathbf{x}^*$, $p(\mathbf{f}, f^*|\mathbf{X}, \mathbf{x}^*, \boldsymbol{\theta}, \theta_n)$ que tiene la siguiente forma

$$p(\mathbf{f}, f^*|\mathbf{X}, \mathbf{x}^*, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\mathbf{0}, \mathbf{C}_{\boldsymbol{\theta},\theta_n \mathrm{C}}) \tag{5.11}$$

La forma de la densidad de probabilidad de f^* condicionada a las observaciones $\mathbf{X}$, la muestra de test $\mathbf{x}^*$, y el vector de las variables latentes $\mathbf{f}$, $p(f^*|\mathbf{x}^*, \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$, se deduce de la misma forma que la expresión (5.4a):

$$p(f^*|\mathbf{x}^*, \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\mathbf{k}^{*\mathrm{T}}_{\boldsymbol{\theta},\theta_n}\ \mathbf{C}^{-1}_{\boldsymbol{\theta},\theta_n}\mathbf{f},\ \ c^{**}_{\boldsymbol{\theta},\theta_n} - \mathbf{k}^{*\mathrm{T}}_{\boldsymbol{\theta}}\ \mathbf{C}^{-1}_{\boldsymbol{\theta},\theta_n}\ \mathbf{k}^{*}_{\boldsymbol{\theta}}) \tag{5.12}$$

En la práctica, las integrales (5.8) y (5.9) son intratables; por tanto, se necesitan soluciones numéricas o aproximaciones analíticas. En el caso binario, la integral (5.9) no es un problema porque es unidimensional y cualquier técnica numérica estándar de integración sirve para calcularla. En el caso de la integral (5.8) existen varios métodos de cálculo; algunos son numéricos, como es el muestreo de MonteCarlo "Markov Chain MonteCarlo" [Neal1993]; otros son aproximaciones analíticas. En las siguientes secciones presentamos brevemente las tres aproximaciones analíticas habitualmente aplicadas: Laplace [Williams1998], EP [Minka2001a], y EM-EP [Kim2006]; consideramos los correspondientes clasificadores (Laplace GP, EP GP, y EM-EP GP) como máquinas de referencia en la parte experimental.

5.2.1. El método de Laplace para GPC

La aproximación de Laplace reemplaza $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ por una densidad gaussiana

$$q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\tilde{\mathbf{f}}, \mathbf{H}^{-1}) \tag{5.13}$$

donde $\tilde{\mathbf{f}}$ es la moda de $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ (el argumento de su valor máximo con respecto a $\mathbf{f}$), y $\mathbf{H}$ es el Hessiano de $-\log p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ en $\tilde{\mathbf{f}}$. Así, la integral (5.8) se aproxima por una convolución de gaussianas. Los valores de $\boldsymbol{\theta}$ y θ_n se determinan maximizando el logaritmo de $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ con respecto a cada hiperparámetro θ (véase el Apéndice D).

Este método es sencillo, pero sufre ciertas limitaciones; por ejemplo, degradación de prestaciones en problemas de alta dimensión [Kuss2005].

5.2.2. La aproximación EP para GPC

El método EP propone la siguiente expresión para $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$:

$$p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) \propto p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)p(\mathbf{t}|\mathbf{f}) = p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) \prod_{k=1}^{K} P(t^{(k)}|f^{(k)}) \tag{5.14}$$

y reemplaza $\{P(t^{(k)}|f^{(k)})\}$ por una aproximación local de la verosimilitud:

$$\tilde{g}_k(f^{(k)}) = s^{(k)} exp(-\frac{(f^{(k)} - m^{(k)})^2}{2v^{(k)}}), \quad \forall k \tag{5.15}$$

donde $\{s^{(k)}, m^{(k)}, v^{(k)}\}$ se determinan iterativamente mediante el algoritmo EP (que está detallado en el Apéndice D). Esto lleva a una aproximación gaussiana de $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ como sigue:

$$q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5.16a}$$

donde

$$\boldsymbol{\Sigma} = (\mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} + \mathbf{V}^{-1})^{-1} \tag{5.16b}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{V}^{-1}\mathbf{m} \tag{5.16c}$$

$\mathbf{V}$ y $\mathbf{m}$ son $diag(v^{(1)}, \ldots, v^{(K)})$ y $[m^{(1)}, \ldots, m^{(K)}]^T$, respectivamente.

Estos clasificadores pueden encontrarse con problemas de convergencia [Minka2001b]; afortunadamente, no han aparecido en nuestros experimentos.

5.2.3. EM-EP para GPC

El método EM-EP modifica el procedimiento anterior incluyendo la estimación de $\boldsymbol{\theta}$ y θ_n aplicando EM:

- Etapa E: se aplica el algoritmo EP;

- Etapa M: se reestiman $\boldsymbol{\theta}$ y θ_n mediante la maximización del límite inferior $F(\boldsymbol{\theta}, \theta_n)$ de la desigualdad de Jensen

$$\begin{aligned} F(\boldsymbol{\theta}, \theta_n) &= \int q_{\mathrm{E}}(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) \log \frac{P(\mathbf{t}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{q_{\mathrm{E}}(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)}\, d\mathbf{f} \\ &\leq \log P(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) \end{aligned} \tag{5.17}$$

donde el subíndice E de la distribución q indica la forma obtenida en la etapa E. Este método suele proporcionar buenas estimaciones de los hiperparámetros pero tiene un alto coste computacional.

Los detalles de esta aproximación, como de las anteriores, se encuentran en el Apéndice D.

5.3. Pruebas experimentales

5.3.1. Conjuntos de datos

Hemos trabajado con ocho problemas de clasificación: contraceptive, crabs, credit, hepatitis, image, ionosfera, pima y ripley. El último problema [Ripley1994] es un problema sintético bidimensional con 8 % de tasa de error bayesiana. Crabs y pima se toman del sitio web PRNN ("Pattern Recognition and Neural Networks") [PRNN]. Los problemas restantes son de la "UCI Machine Learning Repository" [Blake]. Nos referiremos a estos problemas como **con**, **cra**, **cre**, **hep**, **ima**, **ion**, **pim**, y **rip**, respectivamente. La Tabla 5.1 presenta sus principales características.

5.3.2. Diseño de los clasificadores EST-GP

Diseñamos clasificadores tipo GP basados en ESTs, denominados "EST-GP", empleando tres guías auxiliares: un MLP, una máquina GPC, y una SVM, para valorar la importancia de usar diferentes guías auxiliares en el diseño. Nos refe-

Problemas	Entrenamiento (+1/−1)	Test (+1/−1)	Dimensión (D)
con	883 (506/377)	590 (338/252)	9
cra	120 (59/61)	80 (41/39)	7
cre	414 (167/247)	276 (140/136)	15
hep	93 (70/23)	62 (53/9)	19
ima	1848 (821/1027)	462 (169/293)	18
ion	201 (101/100)	150 (124/26)	34
pim	461 (161/300)	307 (107/200)	8
rip	250 (125/125)	1000 (500/500)	2

Tabla 5.1: Principales características de los problemas de clasificación utilizados en la parte experimental.

rimos a los diseños EST-GP correspondientes a dichas guías como EST-GP$_{\text{MLP}}$, EST-GP$_{\text{GPC}}$, y EST-GP$_{\text{SVM}}$, respectivamente[2].

Con respecto al EST-GP$_{\text{MLP}}$, los parámetros libres son el número de neuronas ocultas del MLP auxiliar, N, y los del énfasis μ, α_1, y α_2, que se determinan mediante CV de 10 particiones del conjunto original de entrenamiento (90 % para entrenamiento y 10 % para validación), con 20 repeticiones independientes inicializando aleatoriamente los pesos del MLP, explorando los siguientes valores:

- N: 4, 6, 8, 10 ,12, 14, 16
- μ: 0.01, 0.1, 0.3, 0.6, 1, 1.2, 1.6, 2
- α_1, α_2: 0.001, 0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 3, 4, 5.

Para el EST-GP$_{\text{GPC}}$, se emplea una guía auxiliar GPC con la aproximación

[2]La salida de las guías GPC y SVM se normaliza entre -1 y 1 según la expresión $o_{aux} = z/\,\text{máx}(|z|)$, siendo z y o_{aux} la salida no normalizada y la salida normalizada de la guía auxiliar, respectivamente.

de Laplace. Los parámetros libres del diseño EST-GP$_{\text{GPC}}$, μ, α_1, y α_2 se exploran con la misma CV en los mismos intervalos anteriores del diseño de EST-GP$_{\text{MLP}}$.

Con respecto al EST-GP$_{\text{SVM}}$, sus parámetros libres son C (factor de penalización), σ (dispersión del núcleo gaussiano de la guía auxiliar), μ, α_1, y α_2; que se determinan mediante la misma forma de CV, en que C y σ se exploran en [0.1, 1, 10, 10^2, 10^3, 10^4] y $\sqrt{D} \times$ [2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1, 2, 2^2, 2^3, 2^4, 2^5], respectivamente; y los otros parámetros libres en los intervalos de arriba.

Como máquinas de referencia, consideramos los clasificadores convencionales: un MLP de N' neuronas ocultas y una SVM de parámetros C' y σ'. N', C', y σ' se exploran mediante la misma CV sobre los valores empleados para N, C, y σ, respectivamente.

Las máquinas SVM se entrenan con la "toolboox" IRWLS ("Iteratively Reweighted Least Squares") para SVM [Pérez-Cruz2001], con tolerancia $\epsilon' = 10^{-5}$ para resolver el problema de la programación cuadrática; y los clasificadores GP (Laplace GP, EP GP, EM-EP GP, EST-GP$_{\text{MLP}}$, EST-GP$_{\text{GPC}}$, y EST-GP$_{\text{SVM}}$) con el software GPML ("Gaussian Processes for Machine Learning") [GPML], usando como función núcleo la conocida forma [Kim2006]:

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \theta_0 \exp\{-\frac{1}{2}\sum_{d=1}^{D} l_d(x_d - x'_d)^2\} + \theta_1 + \theta_2 \delta_{\mathbf{x}\,\mathbf{x}'} \tag{5.18}$$

donde $\theta_0 \exp\{-0.5 \ \sum_{d=1}^{D} l_d(x_d - x'_d)^2\}$ es la función de covarianza exponencial cuadrática con determinación automática de relevancia (ARD, "Automatic Relevance Determination"). El hiperparámetro θ_0 es la escala vertical total de la variación de la variable latente f, y l_d es la escala de la longitud de la característica que corresponde a la dimensión d. θ_1 es el hiperparámetro de la función de covarianza "constante". θ_2 es la varianza del ruido blanco de la función de covarianza. La inicialización de estos hiperparámetros se hace del siguiente modo: $l_d = 0.05$ para $d = 1, \ldots, D$; $\theta_0 = 1, \theta_1 = 10^{-3}$; y $\theta_2 = 10^{-4}$. Para conjuntos de datos de tamaño relativamente grande, como **con** e **ima**, se aplica el método disperso de Snelson y Ghahramani, SPGP ("Sparse Pseudo-input Gaussian Processes") [Snelson2006] para reducir la complejidad computacional del entrenamiento de los GPs, del orden de $O(K^3)$, y el coste de la predicción, en la fase

de test, del orden de $O(K^2)$.

5.3.3. Resultados

La Tabla 5.2.**A** presenta los resultados de los experimentos que proporcionan los EST-GP$_{\text{MLP}}$, EST-GP$_{\text{GPC}}$, y EST-GP$_{\text{SVM}}$, comparados con Laplace GP, EP GP, EM-EP GP, y los clasificadores convencionales: el MLP y la SVM. Además, se incluyen en la Tabla 5.2.**B** los resultados de los casos particulares de los diseños EST que corresponden a $\alpha_1 = \alpha_2$, y $\mu = 0$ y $\alpha_1 = \alpha_2$. Denominamos a estos diseños EST-GP$_{\text{1MLP}}$, EST-GP$_{\text{1GPC}}$, EST-GP$_{\text{1SVM}}$, EST-GP$_{\text{2MLP}}$, EST-GP$_{\text{2GPC}}$, y EST-GP$_{\text{2SVM}}$[3].

Sistematizaremos la discusión en varios bloques de comparaciones.

i) Comparación entre los clasificadores GP

La comparación entre los clasificadores estándares GP revela que EM-EP GP ofrece mejores prestaciones que EP GP y Laplace GP para todos los problemas salvo en **ima**; destacando una diferencia importante con respecto a EP GP para **cra** y **hep**. En **ima**, EP GP es claramente mejor que EM-EP GP, que ofrece el peor resultado. En general, Laplace GP ofrece comparativamente las peores prestaciones.

ii) Clasificadores convencionales vs. clasificadores estándares GP

El MLP tiene una clara ventaja con respecto al mejor clasificador GP para todos los casos salvo en **con**, **cre**, **pim**, y **rip**; la diferencia es inapreciable para **pim** y pequeña (a favor del EM-EP GP) para **con**, **cre**, y **rip**. De otra parte, la SVM es mejor que el MLP para **ima** y **ion**, peor (y similar a EM-EP GP) para **hep**, y proporciona casi los mismos resultados para los otros casos. Por tanto, se puede decir que los clasificadores GP no son competitivos con respecto a los clasificadores convencionales. Este hecho nos anticipa que emplear los clasifica-

[3]Para el diseño de EST-GP$_{1\,\text{MLP}}$ para **ima**, se incluye $N = 2$ porque el valor extremo $N = 4$ ofrece el mejor resultado de la CV; y, por similares razones, se consideran $N = 18$ para EST-GP$_{1\,\text{MLP}}$ en **ion**, y $\alpha_1 = 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 30, 35$, para EST-GP$_{2\,\text{MLP}}$ en **ima**.

A	**con**	**cra**	**cre**	**hep**	**ima**	**ion**	**pim**	**rip**
Laplace GP	70.9	90.0	90.2	79.3	79.9	87.3	77.9	90.7
EP GP	71.4	90.0	90.6	83.9	81.8	88.0	77.5	90.6
EM-EP GP	**71.9**	91.3	**90.9**	85.5	76.2	88.7	78.2	**90.8**
MLP CV	70.6±1.4	97.4±0.4	88.3±1.7	89.0±2.8	88.5±2.4	93.3±1.6	78.2±1.5	90.1±0.8
N'	10	10	14	10	6	6	8	14
MLP om*	70.7±1.5	97.4±0.4	88.4±1.1	89.1±2.6	88.5±2.4	93.3±1.6	78.7±1.1	90.3±0.4
N'	6	10	4	4	6	6	16	10
SVM CV	70.8±0.4	97.1±1.2	88.3±0.5	85.8±0.7	**96.4±0.6**	**97.8±0.5**	72.4±1.2	89.9±0.5
C'/σ'	$10^3/2\sqrt{D}$	$10^3/2\sqrt{D}$	$10/2^{-1}\sqrt{D}$	$10/2^{-4}\sqrt{D}$	$10/2^{-3}\sqrt{D}$	$10/2^{-1}\sqrt{D}$	$10^2/2^{-1}\sqrt{D}$	$10^3/2^{-1}\sqrt{D}$
SVM om*	71.3±0.7	99.4±1.2	96.0±0.0	91.9±1.7	96.4±0.6	97.8±0.5	79.8±1.0	91.0±0.5
C'/σ'	$10^2/\sqrt{D}$	$10^2/2^{-1}\sqrt{D}$	$10/2^3\sqrt{D}$	$10/2^{-1}\sqrt{D}$	$10^2/2^{-3}\sqrt{D}$	$10/2^{-1}\sqrt{D}$	$10^2/2^2\sqrt{D}$	$0.1/2^{-4}\sqrt{D}$
EST-GP$_{\text{MLP}}$ CV	**71.9±0.3**	98.4±1.3	88.8±0.8	88.5±3.1	93.6±0.3	95.8±0.7	78.6±0.7	**90.8±0.3**
N_1/μ	12/0.6	12/0.6	6/0.3	14/1	8/1.6	$12/10^{-1}$	12/1.6	10/0.6
α_1/α_2	2/2	$10^{-2}/4$	4/3	$10^{-2}/10^{-2}$	$1/10^{-2}$	$4/10^{-2}$	$10^{-2}/10^{-2}$	0.5 /1
EST-GP$_{\text{MLP}}$ om*	72.1±0.3	98.6±0.7	92.8±1.0	88.8±3.5	93.7±0.3	96.0±0.8	78.9±0.4	90.9±0.3
N_1/μ	16/1.2	10/0.6	$16/10^{-2}$	4/0.6	12/1.6	$12/10^{-2}$	16/1.6	6/1.6
α_1/α_2	0.5/0.1	$10^{-3}/1.5$	$10^{-3}/10^{-2}$	$10^{-3}/1$	1/0.05	$1.5/10^{-2}$	$10^{-3}/0.05$	$0.5/10^{-2}$
EST-GP$_{\text{GPC}}$ CV	71.4±0.9	97.0±1.7	**90.9±1.5**	**91.1±2.4**	89.5±3.2	92.7±1.3	76.9±0.7	90.7±0.3
$\mu/\alpha_1/\alpha_2$	1.2/2/1	0.1/0.05/2	1/0.1/4	0.3/1.5/0.5	0.1/0.05/3	1/1.5/2	1.6/3/1.5	0.3/4/1
EST-GP$_{\text{GPC}}$ om*	71.9±0.6	97.0±1.7	91.1±1.6	92.6±1.6	90.3±2.6	92.9±1.2	77.1±0.5	90.8±0.4
$\mu/\alpha_1/\alpha_2$	0.6/0.1/4	0.1/0.05/2	1/0.1/5	2/1.5/2	2/3/3	1.2/5/3	0.6/1.5/0.1	0.3/1/1.5
EST-GP$_{\text{SVM}}$ CV	71.2±0.5	**99.4±1.2**	90.0±0.8	90.3±2.8	94.5±0.4	96.9±0.9	**79.3±0.6**	90.5±0.5
$C/\sigma/$	$10^3/2\sqrt{D}/$	$10/2^{-2}\sqrt{D}/$	$10/2^3\sqrt{D}/$	$10^2/2\sqrt{D}/$	$10^3/2^{-1}\sqrt{D}/$	$1/2^{-2}\sqrt{D}/$	$1/2^{-1}\sqrt{D}/$	$1/2^2\sqrt{D}/$
$\mu/\alpha_1/\alpha_2$	0.3/4/0.05	1.2/0.5/1.5	0.6/1.5/2	1.2/0.1/0.1	0.3/0.05/0.01	0.1/1/0.01	1.6/0.05/0.1	1.6 /0.5/4
EST-GP$_{\text{SVM}}$ om*	72.4±0.7	99.6±1.3	96.3±0.3	94.2±1.9	95.7±0.2	97.9±0.4	79.8±0.9	91.4±0.5
$C/\sigma/$	$10^4/2^5\sqrt{D}/$	$10^3/2^{-1}\sqrt{D}/$	$10/2^3\sqrt{D}/$	$10/2^{-1}\sqrt{D}/$	$0.1/2^{-2}\sqrt{D}/$	$10/2^{-1}\sqrt{D}/$	$10^2/2^2\sqrt{D}/$	$10^2/2^{-2}\sqrt{D}/$
$\mu/\alpha_1/\alpha_2$	0.1/0.05/2	2/1.5/0.5	2/0.1/0.1	$0.3/0.1/10^{-3}$	1/0.1/1.5	0.1/1/0.1/	0.1/0.05/0.1	1.2/0.05/0.01

B	**con**	**cra**	**cre**	**hep**	**ima**	**ion**	**pim**	**rip**
EST-GP$_{\text{1MLP}}$ CV	**71.9±0.3**	97.2±1.7	88.1±0.9	88.5±3.1	**94.6±0.4**	92.1±1.4	**78.6±0.7**	89.4±2.3
$N/\mu/\alpha_1$	12/0.6/2	6/0.6/0.05	$6/1/10^{-2}$	$14/1/10^{-2}$	2/2/4	16/2/1	$12/1.6/10^{-2}$	6/0.1/0.05
EST-GP$_{\text{1MLP}}$ om*	72.1±0.3	98.1±0.6	92.8±1.0	88.5±3.1	95.0±0.3	95.9±0.8	78.8±1.0	90.8±0.4
$N/\mu/\alpha_1$	16/1/0.5	6/1.2/0.5	$16/10^{-2}/10^{-2}$	$14/1/10^{-2}$	2/1.6/0.5	$12/10^{-2}/10^{-2}$	$16/1/10^{-3}$	10/1.6/0.1
EST-GP$_{\text{2MLP}}$ CV	71.4±1.1	96.7±1.0	88.9±0.8	83.2±3.8	94.3±0.3	93.4±1.3	73.7±1.2	90.4±0.6
N/α_1	6/1	$14/10^{-2}$	8/1.5	16/4	8/25	10/1.5	8/1	14/3
EST-GP$_{\text{2MLP}}$ om*	72.2±0.6	97.0±1.0	89.0±0.8	86.7±4.1	94.3±0.3	93.8±1.2	76.5±0.7	90.5±0.3
N/α_1	6/3	$4/10^{-3}$	6/1.5	$4/10^{-3}$	4/25	8/1	8/5	$16/10^{-3}$
EST-GP$_{\text{1GPC}}$ CV	70.6±1.5	97.0±1.7	89.0±1.2	**90.7±3.6**	87.3±2.7	90.6±2.0	75.7±1.5	**90.7±0.4**
μ/α_1	2/1	0.1/2	0.3/0.5	10^{-2}/1.5	0.1/1	$1/10^{-2}$	0.3/1.5	0.3/2
EST-GP$_{\text{1GPC}}$ om*	71.7±0.6	97.0±1.7	89.7±1.3	92.6±1.6	90.3±2.6	92.9±1.2	76.9±0.6	90.7±0.4
μ/α_1	1/2	0.1/2	1/0.5	2/1.5	2/3	1.2/5	1.2/4	0.3/2
EST-GP$_{\text{2GPC}}$ CV	70.9±0.7	96.1±1.1	90.0±1.0	81.9±3.3	87.8±4.8	90.3±2.7	76.7±0.7	90.5±0.6
α_1	1	3	1	0.1	1	0.5	4	0.1
EST-GP$_{\text{2GPC}}$ om*	71.1±0.9	96.4±1.4	90.1±1.0	84.2±3.7	89.0±4.0	92.3±1.4	77.4±0.5	90.5±0.4
α_1	3	5	0.5	5	4	3	10^{-2}	2
EST-GP$_{\text{1SVM}}$ CV	71.2±0.5	**99.4±1.2**	90.0±0.8	90.3±2.8	94.3±0.2	**95.3±0.9**	76.4±0.8	90.2±0.3
$C/\sigma/$	$10^3/2\sqrt{D}/$	$10/2^{-2}\sqrt{D}/$	$10/2^3\sqrt{D}/$	$10^2/2\sqrt{D}/$	$10^3/2^{-1}\sqrt{D}/$	$1/2^{-2}\sqrt{D}/$	$10^2/\sqrt{D}/$	$10/2^4\sqrt{D}/$
μ/α_1	0.3/0.05	1.2/0.5	0.6/2	1.2/0.1	0.1/1.5	1/0.05	0.3/0.05	1.6/0.5
EST-GP$_{\text{1SVM}}$ om*	72.4±0.7	99.6±1.2	96.3±0.3	93.4±1.6	95.5±0.5	97.9±0.4	79.8±0.9	91.3±0.5
$C/\sigma/$	$10^4/2^5\sqrt{D}/$	$10^3/2^{-1}\sqrt{D}/$	$10/2^3\sqrt{D}/$	$10/2^{-1}\sqrt{D}/$	$10^2/2^{-4}\sqrt{D}/$	$10/2^{-1}\sqrt{D}/$	$10^2/2^2\sqrt{D}/$	$10^2/2^{-2}\sqrt{D}/$
μ/α_1	0.1/2	2/1.5	2/0.1	$0.1/10^{-2}$	1.6/0.5	0.3/0.01	0.1/0.05	1.2/0.05
EST-GP$_{\text{2SVM}}$ CV	71.0±0.5	**99.4±0.9**	**91.4±0.9**	87.1±3.4	93.8±0.7	94.7±1.5	76.4±0.7	90.4±0.5
$C/\sigma/\alpha_1$	$10^3/2\sqrt{D}/0.1$	$10^2/2^{-1}\sqrt{D}/0.5$	$10/2^3\sqrt{D}/3$	$10^3/2\sqrt{D}/1$	$10^3/\sqrt{D}/0.1$	$1/2^{-2}\sqrt{D}/5$	$10^3/2^{-2}\sqrt{D}/3$	$10^3/2^{-2}\sqrt{D}/1$
EST-GP$_{\text{2SVM}}$ om*	72.3±0.4	99.6±1.2	96.0±0.1	91.3±3.0	94.6±0.3	95.7±1.0	79.8±0.9	90.6±0.3
$C/\sigma/\alpha_1$	$10^4/2^5\sqrt{D}/3$	$10^3/2^{-1}\sqrt{D}/0.5$	$10^2/2^5\sqrt{D}/0.1$	$10/2^{-1}\sqrt{D}/1$	$10^4/2^{-4}\sqrt{D}/0.5$	$1/\sqrt{D}/2$	$10^2/2^2\sqrt{D}/10^{-3}$	$1/2^{-1}\sqrt{D}/1$

Tabla 5.2: Tasa de acierto de clasificación (± desviación estándar) de Laplace GP, EP GP, EM-EP GP, MLP, SVM, EST-GP$_{\text{MLP}}$, EST-GP$_{\text{GPC}}$, y EST-GP$_{\text{SVM}}$ (parte **A**), EST-GP$_{\text{1MLP}}$, EST-GP$_{\text{2MLP}}$, EST-GP$_{\text{1GPC}}$, EST-GP$_{\text{2GPC}}$, EST-GP$_{\text{1SVM}}$, y EST-GP$_{\text{2SVM}}$ (parte **B**) con datos de test, indicando los parámetros de diseño. *"om" indica los diseños "omniscientes".

dores GP como máquinas auxiliares para los EST-GPs no debe ofrecer buenos resultados, dada la importancia de la calidad de la máquina auxiliar para los diseños EST.

iii) Diseños EST-GP vs. clasificadores estándares GP y clasificadores convencionales

Con respecto al mejor de los clasificadores GP, los resultados de EST-GP_{MLP} muestran una ventaja significativa para **cra**, **hep**, **ima**, e **ion** -precisamente para estos problemas, el MLP tiene una ventaja con respecto a los clasificadores GPs-, y clara desventaja para **cre**, donde el MLP ofrece las peores prestaciones. Con respecto a las máquinas convencionales (MLP y SVM), el EST-GP_{MLP} supera claramente al MLP en **con**, **cra**, **ima**, e **ion**, y nunca ha ofrecido peores resultados. También, EST-GP_{MLP} mejora la SVM en problemas como **con**, **cra**, **hep**, **pim**, y **rip**, aunque es peor en **ima** e **ion**.

Para el diseño EST-GP_{GPC}, se aprecia que hay mejora con respecto a los clasificadores GP para **con**, **hep** e **ima**, y con respecto al MLP estándar para **con**, **cre**, **hep**, e **ima**; en comparación con la SVM, se obtiene mejora para los problemas **con**, **cre**, **hep**, **pim** y **rip**. Pero, si comparamos los resultados de este diseño con los demás EST-GPs (con guías auxiliares MLP y SVM), se observa que EST-GP_{GPC}, en general, no es competitivo con EST-GP_{MLP} y EST-GP_{SVM} para la mayoría de los problemas (salvo en **cre** y **hep**, donde hay una ligera ventaja), aún siendo guía y máquina final de la misma familia.

Por otro lado, EST-GP_{SVM} lleva una clara ventaja con respecto a los clasificadores GP en los mismos casos que EST-GP_{MLP} (para los que he de resaltarse que la SVM convencional es mejor que el MLP), una ligera ventaja en **pim**, y una pequeña desventaja en **cre**. Además, el EST-GP_{SVM} supera a la SVM en todos los problemas salvo en **ima** e **ion**.

Los resultados que se acaban de mencionar prueban la efectividad de la idea EST para diseñar clasificadores GP de altas prestaciones. Por otro lado, parece clara la importancia de la calidad de la máquina auxiliar para obtener buenos resultados, aunque, si la máquina auxiliar exhibe altas prestaciones y la máquina final EST-GP no ofrece los resultados esperados, probablemente es porque la

estructura y el entrenamiento de la máquina final no son tan adecuados como los de la auxiliar para resolver el problema en cuestión. También se ha verificado, en los experimentos de EST-GP utilizando clasificadores GP como máquinas auxiliares (sin obtener resultados útiles), que la (relativamente) baja calidad de las máquinas auxiliares GP no permite obtener resultados útiles.

iv) Prestaciones globales de los diseños EST-GP

En la Tabla 5.2.**A** se presentan en negrita los mejores resultados. Estos corresponden a los diseños EST-GP_{MLP} para **con** y **rip** (en este caso, comparte la primera posición con EM-EP GP), EST-GP_{GPC} para **cre** (también comparte el mejor resultado con EM-EP GP) y **hep**, y EST-GP_{SVM} para **cra** y **pim**, mientras que la SVM convencional es la mejor en los casos de **ima** e **ion** (destacando que el diseño EST-GP_{SVM} alcanza la segunda posición). Estas comparaciones parecen indicar que las técnicas EST son interesantes para construir clasificadores muy competitivos en cuanto a prestaciones.

v) Diseños EST-GP vs. diseños EST-GP simplificados (correspondientes a los casos particulares)

Como cabía esperar, cuando se reduce el número de los parámetros libres, normalmente las prestaciones de los diseños EST-GP decrecen. Pero hay algunos casos donde esto no ocurre: las prestaciones crecen, probablemente porque el correspondiente proceso de CV es sencillo y fácil.

La Tabla 5.2.**B** muestra los valores de la tasa de acierto de clasificación para las versiones simplificadas tipo 1 y 2 (correspondientes a los casos particulares de los diseños EST-GP $\alpha_1 = \alpha_2$, y $\mu = 0$ y $\alpha_1 = \alpha_2$, respectivamente). Los casos marcados en negrita en la parte **B** de la Tabla 5.2 representan los mejores resultados obtenidos por los diseños simplificados tipo 1 y 2 de los EST-GPs [4]:

- con EST-GP_{1MLP}, para **con** (similar) e **ima** (con mejora).
- con EST-GP_{1GPC}, para **hep** (empeorando) y **rip** (similar).

[4]La comparación se realiza entre los diseños EST-GP simplificados vs. los EST-GPs genéricos.

- con EST-GP_{1SVM}, para **cra** (similar).
- con EST-GP_{2SVM}, para **cra** (similar) y **cre** (con mejora).

Además, hay algunos casos donde los diseños EST-GP tipo 1 se degradan (EST-GP_{MLP} para **cra**, **ion**, y **rip**; EST-GP_{GPC} para **con**, **cre**, **hep**, **ima**, **ion**, y **pim**; EST-GP_{SVM} para **ion** y **pim**), y hay otros casos que muestran degradación para los diseños EST-GP tipo 2 (**hep** y **pim** para EST-GP_{MLP}, pero se recupera para **rip**; **con**, **cra**, **cre**, **hep**, **ima**, e **ion** para EST-GP_{2GPC}; **hep** para EST-GP_{SVM}). Pero debemos resaltar que esto es el coste de reducir la carga computacional de la CV aproximadamente en cantidades correspondientes al número de los valores de los parámetros libres que desaparecen; es decir, se han reducido 11 veces para los diseños tipo 1 y 88 veces para los diseños tipo 2 -dicha reducción corresponde a órdenes de magnitud 1 y 2, respectivamente-.

En general, en el caso de los diseños EST tipo 2, los resultados correspondientes no son en absoluto malos: están entre los mejores para **cra** y **cre** (EST-GP_{2SVM}) y muy próximos al mejor para **con** (EST-GP_{2MLP}). También hay diseños EST simplificados que son mejores que los clasificadores GP estándares, como en **hep** (EST-GP_{2SVM}), **ima** (EST-GP_{2MLP}) e **ion** (EST-GP_{2SVM}). Como quiera que existe una larga serie de diferentes candidatos de funciones de énfasis que sirven para definir ESTs, la mayoría de ellos con un número moderado de parámetros libres, la posibilidad de utilizar la aproximación propuesta parece atractiva y permite elegir entre prestaciones esperadas y carga computacional de los diseños.

5.3.4. Carga computacional

Todos los experimentos anteriores se han llevado a cabo con un "cluster" de computación bastante potente (rendimiento de alrededor de 4 TFlops, 546 núcleos, de 1 a 4 Gbytes de memoria por procesador, ejecución sobre red dedicada de 16 bits/sec, almacenamiento 7 Tbytes centralizados + 7 Tbytes distribuidos, velocidad de los procesadores de 2.6 hasta 3.06 Ghz, 36 equipos con MacOs Leopard y 56 equipos con Linux 84 bits (distribución bentoo), y sistema de gestión de tareas Fura de GridSystems (discontinuado)). Este proceso no permite esti-

mar el esfuerzo computacional de entrenamiento y de operación. Por ello, se ha estimado la carga computacional de acuerdo al procedimiento que se explica a continuación.

	con	**cra**	**cre**	**hep**	**ima**	**ion**	**pim**	**rip**
MLP	2.8(1)	6.4	1.5(1)	7.6	2.1(1)	1.1(1)	1.5(1)	9.7
SVM	2.4(1)	6.0(-2)	9.3(-1)	6.0(-2)	4.0(1)	2.7(-1)	1.0	2.5(-1)
EST-GP$_{\text{MLP}}$	9.1(1)	7.2	2.9(1)	8.7	1.4(2)	1.6(1)	2.7(1)	1.1(1)
EST-GP$_{\text{1MLP}}$	9.1(1)	7.2	2.9(1)	8.7	1.4(2)	1.6(1)	2.7(1)	1.1(1)
EST-GP$_{\text{2MLP}}$	9.1(1)	7.2	2.9(1)	8.7	1.4(2)	1.6(1)	2.7(1)	1.1(1)
EST-GP$_{\text{GPC}}$	1.1(2)	2.7	4.0(1)	3.1	8.5(2)	1.5(1)	9.0(1)	6.2
EST-GP$_{\text{1GPC}}$	1.1(2)	2.7	4.0(1)	3.1	8.5(2)	1.5(1)	9.0(1)	6.2
EST-GP$_{\text{2GPC}}$	1.1(2)	2.7	4.0(1)	3.1	8.5(2)	1.5(1)	9.0(1)	6.2
EST-GP$_{\text{SVM}}$	3.1(1)	8.0(-2)	1.1	9.0(-2)	4.2(1)	3.6(-1)	1.2	2.8(-1)
EST-GP$_{\text{1SVM}}$	3.1	8.0(-2)	1.1	9.0(-2)	4.2(1)	3.6(-1)	1.2	2.8(-1)
EST-GP$_{\text{1SVM}}$	3.1	8.0(-2)	1.1	9.0(-2)	4.2(1)	3.6(-1)	1.2	2.8(-1)

Tabla 5.3: Tiempo (en segundos) de un paso de la CV del MLP convencional, de la SVM, y de los diseños EST: EST-GP$_{\text{MLP}}$, EST-GP$_{1\,\text{MLP}}$, EST-GP$_{2\,\text{MLP}}$, EST-GP$_{\text{GPC}}$, EST-GP$_{1\,\text{GPC}}$, EST-GP$_{2\,\text{GPC}}$, EST-GP$_{\text{SVM}}$, EST-GP$_{1\,\text{SVM}}$, y EST-GP$_{2\,\text{SVM}}$ de los 8 problemas estudiados. ($\bullet$) indica el factor $10^{\bullet}$.

Se mide un promedio (sobre 10 folds) del tiempo de un solo paso de entrenamiento (una realización para un "fold"; véase la Tabla 5.3) de la CV para cada problema usando una máquina que tiene las siguientes características:

- Procesador Turion 64×2 Mobile Technology TL-58 de frecuencia 1.9 GHz;
- Memoria RAM de tamaño 895 MB;
- Disco duro de tamaño 139 GB;

programando los algoritmos en Matlab R2007b. Estas medidas, presentadas en la Tabla 5.3, se multiplican por el número de "folds" (10), de realizaciones (20), y de los parámetros de la CV (7 para N, 6 para C, 11 para σ, 8 para μ, 11 para

α_1, y 11 para α_2; también se consideran los valores adicionales en los casos de exploración extendida) correspondientes al proceso de la CV para cada diseño. El tiempo necesario para el entrenamiento del diseño final se añade a las cantidades resultantes. También se han obtenido los valores correspondientes a los clasificadores GP estándares. Todos los resultados correspondientes a esta estimación del tiempo del diseño para las diferentes máquinas mencionadas en el texto se muestran en la Tabla 5.4.

	con	**cra**	**cre**	**hep**	**ima**	**ion**	**pim**	**rip**
Laplace GP	1.1(2)	1.9	2.5(1)	1.9	8.4(2)	9.2	7.8(1)	4.5
EP GP	1.4(3)	1.0(1)	1.7(2)	5.2	2.9(4)	3.8(1)	2.1(2)	4.5(1)
EM-EP GP	5.0(3)	2.3(1)	8.4(2)	2.2(1)	4.2(4)	4.3(2)	7.3(2)	8.1(2)
MLP	4.5(4)	9.6(3)	2.5(4)	9.6(3)	3.2(4)	1.7(4)	2.4(4)	1.6(4)
SVM	1.6(4)	4.0(1)	6.2(2)	4.3(1)	2.7(4)	1.8(2)	6.7(2)	1.7(2)
EST-GP$_{\text{MLP}}$	1.2(8)	9.5(6)	3.8(7)	9.5(6)	1.9(8)	2.3(7)	3.8(7)	1.6(7)
EST-GP$_{1\,\text{MLP}}$	1.1(7)	8.6(5)	3.5(6)	8.6(5)	2.0(7)	2.4(6)	3.5(6)	1.5(6)
EST-GP$_{2\,\text{MLP}}$	1.4(6)	1.1(5)	4.4(5)	1.3(5)	4.0(6)	2.7(5)	4.4(5)	1.9(5)
EST-GP$_{\text{GPC}}$	1.1(6)	2.6(4)	3.9(5)	3.0(4)	8.3(6)	1.4(5)	8.7(5)	6.0(4)
EST-GP$_{1\text{GPC}}$	1.0(5)	2.4(3)	3.5(4)	2.7(3)	7.5(5)	1.3(4)	7.9(4)	5.5(3)
EST-GP$_{2\text{GPC}}$	1.3(4)	2.2(2)	3.2(3)	2.5(2)	6.8(4)	1.2(3)	7.2(3)	5.0(2)
EST-GP$_{\text{SVM}}$	1.9(7)	5.1(4)	7.2(5)	5.4(4)	2.6(7)	2.4(5)	7.7(5)	1.8(5)
EST-GP$_{1\,\text{SVM}}$	1.8(6)	4.7(3)	6.5(4)	4.9(3)	2.5(6)	2.2(4)	7.0(4)	1.6(4)
EST-GP$_{2\,\text{SVM}}$	2.2(5)	5.8(2)	8.2(3)	6.1(2)	3.1(5)	2.7(3)	8.7(3)	2.0(3)

Tabla 5.4: Tiempo estimado (en segundos) para el diseño de las diferentes máquinas mencionadas en la Tabla 5.2. ($\bullet$) indica el factor $10^{\bullet}$.

De acuerdo con la Tabla 5.4, se puede decir que los diseños EST generales, tipo 1, y tipo 2 necesitan aproximadamente 3, 2, y 1 órdenes de magnitud, respectivamente, más que los diseños correspondientes a las máquinas auxiliares. Estos resultados eran esperables porque corresponden al número de las etapas de la CV que los parámetros ESTs requieren.

Está claro que los diseños basados en MLPs necesitan un importante esfuerzo

computacional: en general, dichos diseños requieren 2 órdenes de magnitud más que el esfuerzo computacional de los diseños basados en máquinas GPC y SVMs. Por consiguiente, utilizar los MLPs como máquinas auxiliares es únicamente razonable cuando los diseños resultantes proporcionan comparativamente buenos resultados.

Con respecto a cuánto aumentan las técnicas EST el esfuerzo necesario para diseñar los clasificadores GP, nos referiremos a los EM-EP GPs para las comparaciones porque, en general, proporcionan los mejores resultados. Respecto a ellos, los diseños EST (basados en máquinas GPC y SVMs) genéricos, tipo 1, y tipo 2 necesitan aproximadamente 3, 2, y 1 órdenes de magnitud más de esfuerzo computacional, respectivamente, que los diseños EM-EP GP. Dadas las claras oportunidades de mejorar sensiblemente las prestaciones, eso no es un precio excesivo; sin embargo, es lo suficientemente importante para adoptar cierta precaución a la hora de aplicar estas ideas: es aconsejable empezar a explorar el efecto de diferentes énfasis sencillos, y posteriormente ir más allá, probando con versiones extendidas de dichas técnicas cuando los resultados sean prometedores, si el problema bajo análisis implica una necesidad real de obtener buenos resultados.

La Tabla 5.5 muestra los tiempos de operación (correspondientes a la misma máquina de cómputo mencionada anteriormente) para clasificar todos los datos de test de cada problema. Nótese que todos los diseños EST que emplean la misma máquina auxiliar necesitan aproximadamente el mismo esfuerzo computacional, y las diferencias menores se deben a las diferencias entre los tamaños de las correspondientes máquinas auxiliares óptimas. Las cargas computacionales de operación de los diseños EST-GP basados en SVMs tienen el mismo orden de magnitud que los de los diseños EST-GP basados en máquinas GPC para todos los problemas salvo para **con** e **ima**, y lo mismo con respecto al clasificador EM-EP GP con la excepción de **ima**, en que los EST-GPs necesitan un tiempo de operación importante. Por lo tanto, desde el punto de vista de la carga de operación, las ventajas de estos diseños EST son gratis. Las máquinas EST-GP basadas en MLPs pagan un orden de magnitud más de carga computacional de operación con respecto a los esquemas basados en máquinas GPC y SVMs.

	con	**cra**	**cre**	**hep**	**ima**	**ion**	**pim**	**rip**
Laplace GP	3.7	1.4(-1)	9.2(-1)	9.0(-2)	2.0(1)	2.8(-1)	8.3(-1)	2.3(-1)
EP GP	2.6(1)	1.3(-1)	3.3	1.3(-1)	2.0(2)	5.6(-1)	3.9	7.8(-1)
EM-EP GP	2.3(1)	1.3(-1)	3.2	1.7(-1)	2.1(2)	5.8(-1)	3.9	7.1(-1)
MLP	3.1	5.8(-1)	3.0	4.0(-1)	2.2(-1)	6.2(-1)	8.4(-1)	6.3
SVM	2.0	2.0(-2)	9.4(-1)	1.0(-1)	2.0	2.0(-1)	6.3(-1)	3.1(-1)
EST-GP$_{\text{MLP}}$	1.5(2)	6.3	5.6(1)	3.1	2.0(2)	2.2(1)	5.3(1)	2.2(1)
EST-GP$_{1\,\text{MLP}}$	1.5(2)	6.3	5.6(1)	3.1	2.0(2)	1.9(1)	5.3(1)	2.2(1)
EST-GP$_{2\,\text{MLP}}$	1.5(2)	6.3	5.6(1)	3.1	2.0(2)	1.9(1)	4.7(1)	2.2(1)
EST-GP$_{\text{GPC}}$	1.6(-1)	3.0(-1)	3.0	3.1(-1)	5.0(-1)	1.1(-1)	2.5	1.3
EST-GP$_{1\text{GPC}}$	1.6(-1)	3.0(-1)	3.0	3.1(-1)	5.0(-1)	1.1(-1)	2.5	1.3
EST-GP$_{2\text{GPC}}$	1.6(-1)	3.0(-1)	3.0	3.1(-1)	5.0(-1)	1.1(-1)	2.5	1.3
EST-GP$_{\text{SVM}}$	1.0(1)	3.1(-1)	2.5	1.6(-1)	1.6(1)	7.8(-1)	2.2	1.3
EST-GP$_{1\,\text{SVM}}$	1.0(1)	3.1(-1)	2.5	1.6(-1)	1.6(1)	7.8(-1)	2.2	1.3
EST-GP$_{2\,\text{SVM}}$	1.0(1)	1.6(-1)	2.5	1.6(-1)	1.6(1)	7.8(-1)	2.5	1.1

Tabla 5.5: Tiempo de clasificación (en segundos) para datos de test de los problemas bajo análisis usando las máquinas mencionadas en la Tabla 5.2. ($\bullet$) indica el factor $10^{\bullet}$.

Mostremos algunos datos adicionales para ayudar a entender las ventajas potenciales de usar diseños EST con CV. De [Gómez-Verdejo2008] y de otros trabajos internos tomamos los siguientes resultados del diseño optimizado del Real Adaboost usando "pequeños" MLPs como aprendices. Los datos son la tasa de acierto de clasificación, el número de aprendices y el número de neuronas ocultas de los aprendices:

-**con**: 71.0; 33.7; 2

-**cra**: 97.5; 11.1; 2

-**hep**: 91.4; 10.6; 17

-**ima**: 97.5; 19.6; 11

-**ion**: 95.1; 13.4; 5

-**rip**: 90.3; 28.9; 48

Nótese que todos estos conjuntos de máquinas tienen un gran número de aprendices MLP, mientras que sus prestaciones son mejores que los esquemas ESTs únicamente para **hep** e **ima**.

5.3.5. Sensibilidad con respecto a los parámetros de los diseños por CV

La aplicación de la CV no sólo aumenta la carga computacional para el diseño de las máquinas, sino que introduce un riesgo implícito de seleccionar valores inadecuados para los parámetros de diseño. Un estudio exhaustivo de la sensibilidad con respecto a los parámetros a fijar por CV es complicado; en consecuencia, analizaremos sólo las partes más relevantes (en todo caso, véanse en el Apéndice B las tablas de sensibilidad de estos diseños con respecto a los parámetros libres fijados por la CV correspondientes a todos los problemas considerados en la parte experimental).

En primer lugar, nótese que es una buena señal, con respecto a los parámetros de los diseños EST, que los diseños EST tipo 1 y 2 no muestren una extrema degradación cuando los comparamos con los diseños EST generales. Sin embargo, esto no es suficiente, y no dice nada sobre los otros parámetros de diseño. Por tanto, consideramos aquí un método sencillo para realizar una primera evaluación de la sensibilidad, empleando los diseños llamados "omniscientes".

Los diseños omniscientes son los obtenidos mediante la selección de los parámetros libres de acuerdo a las prestaciones medidas con los datos de test (de acuerdo con la mejor tasa de acierto de clasificación). Obviamente, no son diseños válidos; pero proporcionan una indicación de la sensibilidad de los diseños obtenidos por CV con respecto a los valores de sus parámetros libres. En primer lugar, se comparan las prestaciones de los diseños omniscientes y de los diseños CV para observar la degradación que el proceso de CV produce -si la degradación es moderada, el diseño CV es aceptable-; en segundo lugar, se observan las diferencias entre los valores de los parámetros libres de los diseños omniscientes y los de los

diseños CV, que indican cuán eficaz es la CV para cada uno de estos parámetros.

Incluimos los datos (prestaciones y valores de los parámetros de diseño) correspondientes a los diseños omniscientes en la Tabla 5.2. De acuerdo con esta tabla, la diferencia entre el omnisciente y la CV de los diseños ESTs generales es relevante:

- para los EST-GP_{MLP}s,en **cre** -prestaciones- y en **cre**, **hep**, y **rip** -valores de los parámetros-;

- para los EST-GP_{GPC}s, en **hep** -prestaciones- y en **con**, **hep**, **ima**, **ion**, y **pim** -valores de los parámetros-;

- para los EST-GP_{SVM}s, en **cre** y **hep** -prestaciones- y en **con**, **cra**, **cre**, **ima**, **pim**, y **rip** -valores de los parámetros-.

De acuerdo con lo anterior, se puede decir que el proceso de CV es eficaz para obtener buenos diseños, pero las dificultades son comparativamente altas para los diseños EST basados en SVMs. Nada sorprendente, porque la sensibilidad de las SVMs con respecto a los valores de los parámetros libres fijados por CV es relativamente alta.

Si dibujamos las curvas de prestaciones de un diseño alrededor de cada valor de los parámetros libres obtenidos por CV, podemos tener una idea clara sobre la sensibilidad. Consideramos el caso de **cre**, que parece presentar altas diferencias.

La Figura 5.1, referida al EST-GP_{MLP}, revela que las curvas correspondientes a N, α_1, y α_2 son casi planas, y que la dificultad para EST-GP_{MLP} se debe principalmente a la sensibilidad con respecto al parámetro μ (la curva correspondiente presenta altas variaciones para valores grandes de μ); pero en ningún caso este hecho produce efectos dramáticos.

Según la Figura 5.2, se percibe una sensibilidad moderada del diseño EST-GP_{GPC} con respecto a todos sus parámetros libres, aunque el punto (diamante) correspondiente al diseño obtenido por CV siempre está en las zonas de altas prestaciones. Con respecto a μ, se percibe que para valores superiores a 1.2, las prestaciones del diseño EST-GP_{GPC} se degradan, y lo mismo se puede decir con

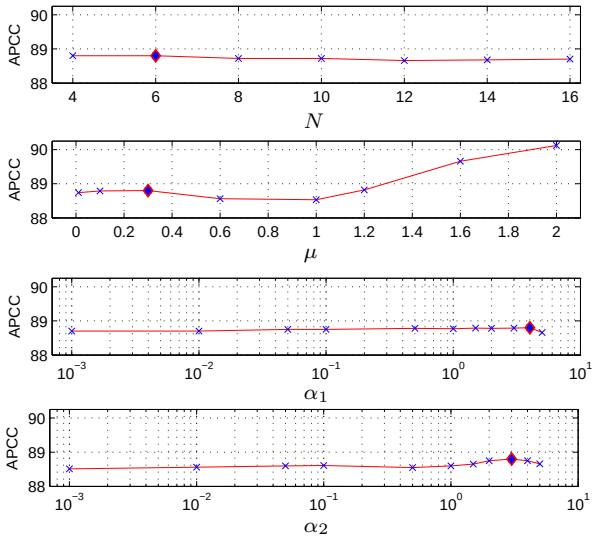

Figura 5.1: Sensibilidad del diseño EST-GP$_{\text{MLP}}$ con respecto a los parámetros libres N, μ, α_1, y α_2 para **cre**. APCC ("Average Percentages of Correct Classification") es el porcentaje de acierto de clasificación. Los diamantes indican los valores de los parámetros del diseño CV.

respecto al parámetro α_1, para el que se observa una ligera degradación en la tasa de acierto de clasificación para valores grandes; con respecto a la curva de α_2, la tasa de acierto se deteriora para valores de α_2 inferiores a 1. Es decir, los diseños con máquinas GPC como guías, además de ser poco competitivos, muestran alta sensibilidad en CV.

Finalmente, la Figura 5.3 ilustra la sensibilidad del diseño EST-GP$_{\text{SVM}}$ con respecto a sus parámetros libres C, σ, μ, α_1, y α_2, en el problema **cre**. Se percibe que el EST-GP$_{\text{SVM}}$ presenta una alta sensibilidad con respecto a μ y α_2. Con respecto a C y α_1, las curvas son casi planas salvo en las regiones correspondientes a pequeños valores de C y α_1, donde las prestaciones se degradan. Por otro lado, las curvas correspondientes a σ y α_2 decrecen para valores superiores a 50 y 0.8, respectivamente. El punto óptimo de la CV siempre está en las regiones de altas prestaciones con respecto a todos los parámetros, salvo μ y α_2. Por lo tanto, se percibe una alta sensibilidad del diseño EST-GP$_{\text{SVM}}$ con respecto a todos los parámetros libres. Como se ha observado en las tres últimas figuras que el problema **cre** es un caso particularmente delicado con respecto a

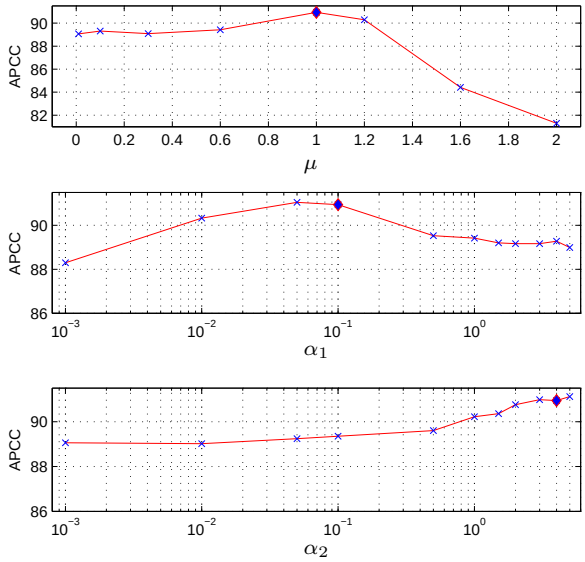

Figura 5.2: Sensibilidad del diseño EST-GP_{GPC} con respecto a los parámetros libres μ, α_1, y α_2 para **cre**. Los diamantes indican los valores de los parámetros del diseño CV.

la CV -nótese que el SVM convencional encuentra dificultades-, se requerirían para él procedimientos más eficientes -y computacionalmente más costosos- para determinar los valores adecuados de los parámetros no entrenables si se desea obtener una máquina potente, como por ejemplo, una CV densa o evaluaciones LOO ("Leave-One-Out").

Las curvas de sensibilidad correspondientes a los demás problemas (véanse sus tablas en el Apéndice B) son consistentes con los resultados obtenidos a partir de la comparación de los resultados omniscientes. Por consiguiente, podemos repetir que, en general, la aplicación de las técnicas EST para el diseño de los clasificadores GP presenta una sensibilidad moderada con respecto a los parámetros no entrenables.

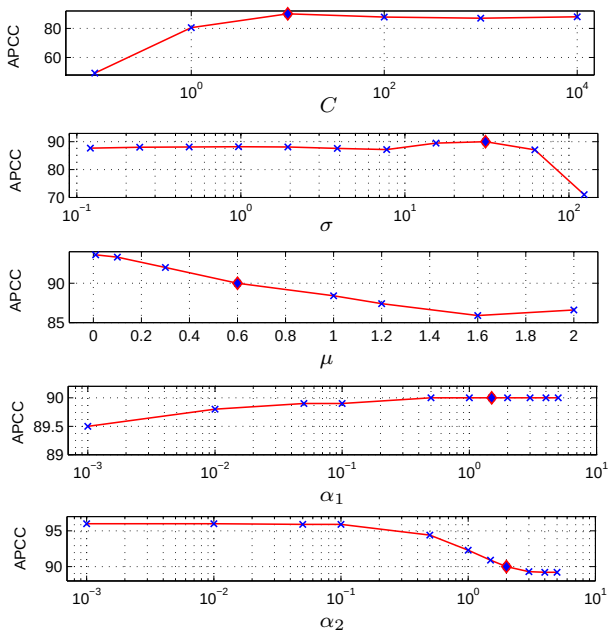

Figura 5.3: Sensibilidad del diseño EST-GP$_{\text{SVM}}$ con respecto a los parámetros libres C, σ, μ, α_1, y α_2 para **cre**. Los diamantes indican los valores de los parámetros del diseño CV.

5.4. Conclusiones

El caso de los GPs resulta particularmente apropiado para la aplicación de las técnicas EST: su formulación para estimación es sencilla, permite una interpretación no muy compleja, proporciona ventajas en generalización, y da una indicación de la calidad de los resultados; pero su uso para clasificación requiere modificaciones trabajosas y no muy productivas, ya que los blancos discretos no encajan con el correspondiente modelado.

En este Capítulo se ha comprobado experimentalmente, sobre un razonable conjunto de problemas de referencia, que la aplicación del potente énfasis anteriormente presentado en la Tesis permite, a costa de un no despreciable esfuerzo computacional, obtener diseños de clasificadores claramente competitivos en cuanto a prestaciones; sin que los problemas de sensibilidad derivados del empleo de CV para determinar los valores de los parámetros no entrenables sean especialmente graves. Incluso versiones simplificados del énfasis -con menor número

de parámetros y, consiguientemente, con menor carga computacional de diseño y reducidos problemas de sensibilidad- resultan ocasionalmente ventajosas: lo que abre la vía a aplicar, en la práctica, aproximaciones graduales para diseñar el tipo de máquinas propuesto: comenzar por énfasis sencillos, y ampliarlos cuando los resultados sean prometedores (aparte, desde luego, de cuando el problema reclame un diseño de altas prestaciones).

Debe destacarse además que, en general, los diseños de las que hemos denominado máquinas EST-GP no requieren carga de operación sensiblemente mayor que los clasificadores GP convencionales.

Por todo ello, cabe concluir que el empleo de los métodos EST para el diseño de clasificadores GP es una opción atractiva que conviene tener en cuenta.

Capítulo 6

Ponderación de muestras vs. ESTs

6.1. Relación entre ponderación de muestras y ESTs

Al avanzar en los trabajos de esta Tesis Doctoral, la diferencia entre los procedimientos que utilizan ponderaciones de muestras de formas generales para entrenar máquinas de clasificación y el concepto de los ESTs ha ido variando de apariencia, hasta que, justamente en el momento en que se alcanzaba el objetivo principal -entrenar clasificadores tipo GP utilizando algoritmos convencionales (de regresión) para dichas máquinas, gracias a la introducción de los ESTs-, las similitudes y las diferencias se apreciaron de forma más nítida. La conexión funcional que existe entre ponderaciones y ESTs fué incluida, junto con una breve discusión, en [El Jelali2011]; pero parece mostrar aún más facetas.

Por lo anterior, se ha decidido incluir este breve capítulo para mostrar y discutir más extensamente esa relación funcional, ya que arroja luz sobre algunos aspectos no totalmente claros de la teoría de las Máquinas de Decisión -particularmente sobre las funciones de activación-, y, a partir de ello, abre la puerta a posibilidades adicionales a las aquí aplicadas.

6.2. Equivalencia funcional

Supondremos por comodidad de notación que la ponderación (no negativa) que, tras obtenerse de información previa -normalmente, de la consideración de los resultados de un clasificador explícito o implícito-, se aplica a la muestra $\mathbf{x}^{(k)}$ para entrenar un clasificador es $(w^{(k)})^2$, y que el clasificador entrega su salida aplicando una activación f a un resultado previo $z^{(k)}$. Si el criterio aplicado es minimizar el error cuadrático[1], el término correspondiente a dicha muestra puede manipularse como sigue:

$$(w^{(k)})^2(t^{(k)} - f(z^{(k)}))^2 = (w^{(k)}t^{(k)} - w^{(k)}f(z^{(k)}))^2 \tag{6.1a}$$

que, obviamente, con las denominaciones

$$t_s^{(k)} = w^{(k)}t^{(k)} \tag{6.1b}$$

$$o^{(k)} = w^{(k)}f(z^{(k)}) \tag{6.1c}$$

pasa a ser

$$(w^{(k)})^2(t^{(k)} - f(z^{(k)}))^2 = (t_s^{(k)} - o^{(k)})^2 \tag{6.2}$$

es decir, el término k del error cuadrático correspondiente al empleo de un "EST" ($t_s^{(k)}$) como referencia para una salida $o^{(k)}$.

Esa salida $o^{(k)}$ es, mientras se mantenga estrictamente la equivalencia, una forma sigmoidal con un amplitud $w^{(k)}$. Pero, si los valores de $\{w^{(k)}\}$ son suficientemente densos, el conjunto de objetivos $\{t_s^{(k)}\}$ toma valores prácticamente continuos (y no ± 1); de modo que permitir que $\{o^{(k)}\}$ pueda tomar valores continuos no parece que vaya a producir efectos tan deletéreos como los que apareja prescindir de la activación cuando los blancos son binarios. Es bien cierto que, en todo caso, no debería permitirse que $|o^{(k)}|$ tomase valores superiores a $w^{(k)}$; pero no puede decirse que no exigirlo sea inaceptable, porque flexibilizar el requisito puede llevar a un mejor aprovechamiento del poder expresivo de la máquina, y, por tanto, cabe suponer -y así lo demuestran los experimentos del Capítulo 3 de esta Tesis- que se obtendrán deterioro o mejora, en principio, según el problema de que se trate.

[1] Lo expuesto se generaliza sin dificultad para costes que sean función del error de salida.

De lo anterior nace la posibilidad de la aplicación directa de los ESTs a la clasificación mediante estimadores tipo GP, cuya formulación ni siquiera admitiría la truncación de los picos de las salidas fuera de un cierto margen.

6.3. Nuevas posibilidades

Dando por bueno el anterior punto de partida, además de la posibilidad de emplear ESTs para el diseño de los clasificadores mediante máquinas de regresión, se abre la posibilidad inversa: diseñar esquemas de ponderación de muestras a partir de la concepción de formas para ESTs, tal y como es la propuesta en esta Tesis (fórmulas (2.7) y (2.8)), cuya potencialidad se ha comprobado en numerosos experimentos.

Tal potencialidad se debe a lo elaborado de esa forma de ESTs: una combinación local convexa que tiene en cuenta el blanco original y la información que proporciona una (buena) máquina de clasificación utilizada como guía, en cuanto a la separación de la frontera y el error, además de la propia salida de esa máquina guía. Naturalmente, así se abre la vía para aprovechar buenos diseños de este tipo para convertirlos en esquemas de ponderación: las posibilidades son casi ilimitadas -elección de la (oportuna) máquina guía, modos de tener en cuenta el error y la separación a la frontera, etc.-

Cabe objetar que el modelo para ESTs propuesto puede dar lugar a ponderaciones negativas. Aunque tal eventualidad parezca preocupante, en realidad no lo es: sólo ocurrirá así cuando la parametrización correspondiente proporcione ventaja (según la CV, desde luego), y lo que entonces se manifiesta es que, para las muestras afectadas por una ponderación negativa, puede resultar hasta conveniente que el tamaño del error de salida crezca: posiblemente solicitar lo contrario supone malversar parte de la capacidad de la máquina que se está utilizando. En cualquier caso, siempre queda al alcance del diseñador prescindir de las muestras cuya ponderación resulte negativa en el proceso de entrenamiento subsiguiente.

Además, hay otras muchas formas de conseguir ESTs y ponderaciones suficientemente generales y que no presenten esa (supuesta) inconsistencia; por

ejemplo, la forma

$$t_s = t\,\{\lambda + (1-\lambda)[\mu\, e^2/4 + (1-\mu)|o|]\} \tag{6.3}$$

con $0 \leq \lambda, \mu \leq 1$, es suficientemente flexible y equivale a una ponderación siempre no negativa.

Nótese que esta vía puede aplicarse para mejorar las prestaciones de clasificadores cuya formulación no obedece directamente al propósito de minimizar un coste convencional, como ocurre con los derivados de aplicar el principio de Máximo Margen (clasificadores SVM y familias emergentes de ellos): cabe ponderar las variables vagas correspondientes a cada muestra, y la consideración de esquemas de ponderación sistemáticos y potentes como los aquí tratados permitiría extraer aún más beneficio de las capacidades de dichas máquinas. Incluso puede pensarse, dado el hecho comprobado de que mejores guías conducen a mejores resultados, en aplicar reiteradamente el proceso, hasta llegar a diseños que muestren efectos de saturación.

Tampoco debe descartarse otra interesante posibilidad: como quiera que existe una relación clara entre ponderaciones y ESTs, no hay obstáculo para convertir unas en otros cuando resulte conveniente: en particular, las sucesivas etapas de entrenamiento de aprendices en conjuntos "Real Adaboost", de manera tal que, al concluir el diseño, todos esos aprendices se puedan integrar, eliminando la capa oculta adicional que se genera cuando se emplean aprendices que incluyen activaciones de salida (ya que las salidas de dichas activaciones han de multiplicarse por los $\{\alpha_t\}$ y después sumarse). No hay riesgo de fracaso por el hecho de que los aprendices sin activación puedan ser menos "fuertes" que sus contrapartes con activación: los aprendices necesarios pueden y deben ser "débiles" (basta con que resulten mejores que el puro azar). La conveniente graduación de la "debilidad" puede conseguirse mediante una oportuna elección de las arquitecturas de los aprendices.

Igual que se ha hecho referencia al "Real Adaboost", se podría haber considerado cualquier otro énfasis adecuadamente elegido (ya demostró Breiman [Breiman1998] que lo fundamental era el remuestreo). Así, en conexión con lo anterior, queda abierta la posibilidad de seleccionar una adecuada forma pa-

ramétrica de ponderación (ESTs) de elevada potencia y proceder de modo análogo a como se hizo en [Gómez-Verdejo2006, Gómez-Verdejo2008] con un simple énfasis mixto uniparamétrico; ofreciéndose así todas las ventajas de adaptar el diseño del conjunto a las características de cada problema, al tiempo que la de integrar todos los aprendices sin necesidad de una capa oculta sobreañadida.

Para concluir este apartado, conviene decir que la observación de la analogía entre los ESTs y las ponderaciones sugiere una posibilidad más: revisitar los clasificadores del tipo GP con el propósito de incorporar a sus formulaciones actuales (Laplace, EP, EP-EM,...), implícita o explícitamente, procesos de ponderación que lleven a mejorar sus prestaciones. Si bien podría buscarse ponderar dentro de los algoritmos propios de estos clasificadores, una opción que verosímilmente resultaría atractiva, por lo inmediato, sería ponderar (de modo complementario al habitual) el nivel del ruido asociado a cada muestra en el propio proceso de estimación mediante GP[2].

6.4. Otra visión de las activaciones

En las secciones anteriores se ha discutido la análoga función que cabe atribuir a las activaciones y a los ESTs en las máquinas de decisión o clasificación, deduciendo además una serie de lineas de trabajo potencialmente beneficiosas. Aunque dicha visión puede resultar chocante a primera vista, deja de serlo en el mismo momento en que se reflexiona sobre la forma sigmoidal precisa para las activaciones: en realidad, aplicarlas supone comprimir la entrada cuando ésta adquiere valores absolutos muy elevados; es decir, reducir en ese caso el valor absoluto del error; y tal cose acontece cuando la muestra correspondiente está claramente bien o claramente mal clasificada. De modo que se le está dando mayor "peso" a las muestras que se encuentran en un entorno de la frontera.

Obviamente, las sigmoides habituales, como la tangente hiperbólica, operan así; y, dado que no puede alegarse para todos los problemas la razón "ideal" de su empleo -se trata de la no linealidad óptima cuando las densidades de probabi-

[2]Nótese que existe una semejanza con los conocidos como modelos heterocedásticos.

lidad de las muestras bajo cada hipótesis son gaussianas con iguales matrices de covarianza-, ha de admitirse que es el comportamiento cualitativo recién descrito lo que hace conveniente su uso[3].

Una vez vista la función de una activación convencional como semejante a una ponderación más intensa de las muestras más próximas a la frontera, cabe preguntarse si no serán posibles elecciones más ventajosas, puesto que se sabe que, dependiendo del problema, puede ser mejor una ponderación que tenga "equilibradamente" en cuenta la importancia de las muestras cercanas a la frontera y las más erróneas -con la verosímil excepción, entre estas últimas, de aquéllas que resulten decididamente situadas en un lugar del espacio de observación que no haga factible pensar en clasificarlas correctamente si no se desea deteriorar la capacidad de generalización del diseño obtenido-. También está muy claro ahora que leves modificaciones de las formas habituales de las activaciones prometen resultar ventajosas para las prestaciones de los clasificadores que las adopten.

Así, por poner un ejemplo sencillo, no parece que haya de ser siempre conveniente que las muestras mal clasificadas den lugar a un error cuyo valor absoluto tiende a 2 precisamente al mismo "ritmo" al que las muestras bien clasificadas producen un error que tiende a 0: no resulta inapropiado pensar que conceder más atención relativa a las segundas que a las primeras puede producir beneficio. Es evidente que tal asimetría de tratamiento puede conseguirse incluso mediante modificaciones elementales: así, incorporar una activación asimétrica de la forma

$$f(z) = \begin{cases} \tanh(z), & \text{si} \quad |e| < 1, \\ \\ \tanh(\alpha\, z), & \text{si} \quad |e| > 1. \end{cases} \tag{6.4}$$

con $0 < \alpha < 1$, supone que para valores de z de signo inadecuado (muestras erróneas: $|e| > 1$), la reducción del error se efectúa menos rápidamente que para los correctamente clasificados ($|e| < 1$).

No menos relevante es la transparencia de la tangente hiperbólica a un solo parámetro de pendiente: $\tanh(\alpha z) = \tanh z'$ si $z' = \alpha z$; con la entrada construida

[3]Aparte de ello, la tangente hiperbólica resulta computacionalmente cómoda porque su derivada es igual a 1 menos el cuadrado de su valor.

mediante un combinador lineal, los pesos ajustarán su nivel hasta llegar a situaciones equivalentes, que corresponderán al menor error cuadrático de salida para un perfil de activación justo de la forma tangente hiperbólica... que no tiene por qué ser el más apropiado para el problema que se pretende resolver. Resulta, por tanto, atractiva la posibilidad de recurrir a activaciones que no muestren esta propiedad de escalado. Una muy sencilla es la falsa potencial

$$f(z) = |z|^{\alpha} sgn\, z \tag{6.5}$$

con $0 < \alpha < 1$, que presenta la forma que muestra la Figura 6.1.

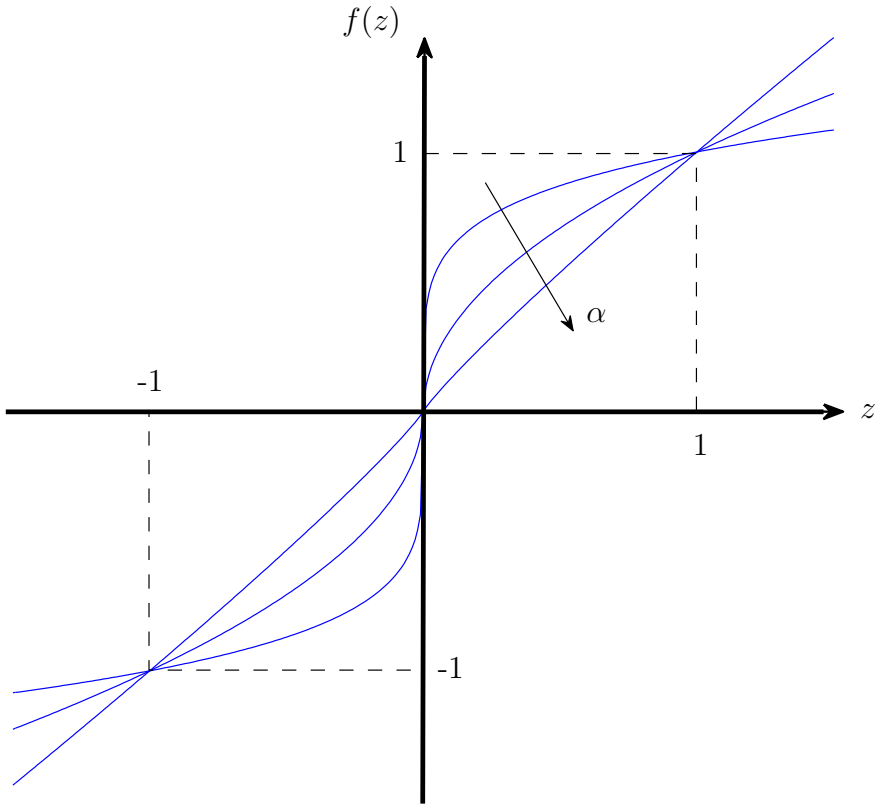

Figura 6.1: Aspecto de la falsa potencial $f(z) = |z|^{\alpha} sgn\, z$, $0 < \alpha < 1$.

Como se puede observar, $\alpha \to 0$ lleva al escalón abrupto, mientras que $\alpha \to 1$ anula la activación; pero en ningún caso puede despreciarse el efecto del nivel de

la entrada. También puede verse que no se produce saturación (± 1 no son valores asintóticos), pero el efecto de esta imperfección no tiene que resultar apreciable si se elige (vía CV) un valor de α apropiado (y, en todo caso, siempre podría forzarse planicidad a partir de $|z| = 1$).

Conviene aquí señalar que, aunque se ha estado suponiendo que α es un parámetro seleccionable y no entrenable para estas activaciones, que se están suponiendo presentes a la salida de la máquina de clasificación, no parece haber obstáculo insalvable para tratarlo como entrenable si se encuentra en capas ocultas; lo que probablemente redundará en conseguir una mayor riqueza expresiva de los diseños resultantes y, con ello, en un mayor compacidad[4]. Incluso puede pensarse en procedimientos adaptativos para aplicar a una activación de salida paramétrica que coronase un combinador lineal.

De cierto que hay muchas alternativas para construir activaciones no lineales flexibles mediante la oportuna introducción de parámetros: será posible tanta más flexibilidad cuantos más parámetros, y mejor, se empleen; apareciendo de nuevo el ya familiar compromiso entre dicha flexibilidad y los requerimientos computacionales para asignar valores a dichos parámetros mediante CV -supuesto que se vuelve a prestar atención a la activación de salida-. Pero no es propósito de este trabajo revisar lo ya aparecido en la literatura técnica y, a partir de esa revisión, proponer y evaluar activaciones parametrizadas: sólo se deseaba destacar la interpretación de las activaciones como elementos reguladores de la importancia de los errores de salida y que esta perspectiva favorece la exploración del ámbito que se está mencionando.

6.5. Conclusiones

En este capítulo, y mediante una adecuada reescritura del coste ponderado para un problema de decisión, se ha puesto de manifiesto y se ha discutido la

[4]Desde un punto de vista estrictamente formal, se haría necesario probar la condición de aproximadores universales para las correspondientes arquitecturas; pero éste es tema que excede el ámbito de la presente Tesis.

semejanza funcional de la ponderación y de la introducción de los ESTs. De esa aparición emergen una serie de oportunidades para introducir ulteriores mejoras en los diseños de clasificador máquina; en particular y de forma resumida:

- cabe transformar en formulaciones ESTs esquemas de ponderación adecuados (suficientemente continuos) y de prestaciones acreditamente buenas (al menos para determinados problemas), para poder aplicar directamente la versión EST a estructuras que corresponden a planteamientos de regresión (como los GMMs o los GPs);

- en sentido contrario, cabe transformar una formulación EST en un esquema de ponderación para aplicarlo, directa o indirectamente, a máquinas de clasificación; incluyendo las que se diseñan siguiendo formulaciones de Máximo Margen;

- las transformaciones en el primer sentido ofrecen la posibilidad de ser empleadas en la construcción de conjuntos vía "boosting" (o "arcing", supuesto que se opta por un énfasis distinto del convencional) para dar lugar a arquitecturas en que desaparece la última capa oculta (por encima de los aprendices) propia de estos procedimientos;

- pueden contemplarse las activaciones de salida como elementos cuyo propósito es también regular la importancia relativa de los errores correspondientes a las diferentes muestras, representando esta perspectiva una guía valida para la apropiada elección de funciones paramétricas que permitan un buen compromiso entre flexibilidad de la (seudo)ponderación implícita y las limitaciones que impone el uso de la CV para determinar los valores de los parámetros de la activación;

- tales activaciones parametrizadas podrían ser también útiles en capas ocultas de máquinas con aprendizaje, cabiendo la posibilidad de entrenar los parámetros de esas activaciones junto con los propios de la máquina convencional.

Capítulo 7

Conclusiones

7.1. Aportaciones de la Tesis

Los trabajos desarrollados a partir de la idea de convertir directamente los blancos duros propios de problemas de clasificación en blancos blandos enfatizados mediante la consideración de lo que ocurre en un clasificador auxiliar o guía, encaminados sobre todo a poder utilizar directamente formulaciones que, como la de los GPs, corresponden intrínsecamente a planteamientos de regresión, han ofrecido los siguientes resultados:

1. La verificación de que los ESTs aplicados a estructuras convencionales (tipo MLP, p.ej.) proporcionan resultados satisfactorias; y que la buena selección de la guía y una suficientemente flexible forma paramétrica para los ESTs llevan a alcanzar prestaciones muy competitivas.

2. En particular, se ha propuesto y aplicado sistemáticamente una elaborada versión de ESTs -fórmulas (2.7) y (2.8)- que auna una notable flexibilidad, debida no sólo a la presencia de parámetros, sino a una conveniente utilización de la información de la guía -combinación convexa local de blanco original y salida de la guía, según el error y la proximidad a la frontera-, con una demanda computacional de CV que no cabe calificar de excesiva; obteniendo excelentes resultados en todas sus aplicaciones prácticas sobre

diferentes esquemas.

3. Se ha comprobado sobre formulaciones de GMMs para regresión y, sobre todo, de GPs, que, efectivamente, el empleo de los ESTs (en la versión indicada en el punto anterior) brinda la posibilidad de llevar a cabo clasificación con, como se ha dicho, excelentes resultados; sin que sea preciso, en el caso de los GPs, recurrir a las modificaciones de su formulación básica que permiten su empleo como clasificadores. Si bien la CV de la versión general supone sensibles incrementos de la carga computacional, los diseños resultantes ofrecen a cambio, en muchas ocasiones, prestaciones superiores a los clasificadores analíticos de tipo GP. Por otro lado, versiones simplificadas de la forma general de ESTs que se ha venido aplicando -con menos parámetros y, consiguientemente, menos demandas computacionales- han demostrado que pueden ocasionalmente seguir siendo ventajosas en cuanto a prestaciones.

4. Como consecuencia del desarrollo de todos los trabajos anteriores y de la reflexión sobre ellos, se ha determinado formalmente una relación de semejanza funcional entre métodos convencionales de ponderación de errores muestrales para clasificación y las formulaciones ESTs; lo que implica unas posibilidades de intercambio que se han discutido en detalle en el Capítulo 6, junto con algunas implicaciones prácticas que, por su atractivo y relevancia, repetiremos más adelante como lineas futuras de actividad. En esa misma discusión se incluye la visión de las activaciones de salida como dispositivos que permiten regular la importancia relativa de los diversos errores, abriendo también interesantes caminos, de los que algunos serán asimismo recordados unas lineas más abajo.

7.2. Sugerencias de futuras lineas de trabajo

1. Convertir formas de ESTs en ponderaciones equivalentes que, por su potencia expresiva, puedan resultar útiles en general; y muy en particular para esquemas de clasificación cuyo entrenamiento no se basa directamente en

la consideración de un coste, como ocurre con los diseños bajo los planteamientos de Máximo Margen (SVMs y familias asociadas), para los que es posible aplicar la ponderación sobre las variables vagas, e incluso apoyarse en ella para llevar a cabo selección de elementos.

2. Transformar la ponderación propia del "Real Adaboost" en una forma de ESTs, a fin de eliminar la capa oculta añadida por este proceso de construcción de conjuntos. Aún más: cabe emplear ya no sólo otras ponderaciones, sino otras formas de ESTs, cuyos parámetros pueden determinarse aprendiz a aprendiz recurriendo a parámetros distintos de los errores -como el de separación-; con lo que, manteniendo la eliminación de la última capa oculta, es previsible que puedan obtenerse diseños de muy altas prestaciones[1].

3. Aplicar iterativamente los procedimientos de tipo ESTs (y también las ponderaciones derivadas de una guía), dado que la calidad de los resultados parece depender de la calidad de la guía, hasta que se manifiesten efectos de saturación en las prestaciones de los consecutivos diseños.

4. Concebir, desde la perspectiva de su efecto sobre los diversos errores, nuevas formas de activaciones de salida que incluyan parámetros ajustables, con el propósito de determinar, con una visión clara de lo que está ocurriendo, cuáles son las formas de activación más adecuadas para la buena resolución de problemas concretos de decisión y clasificación.

5. Finalmente, no es de menor importancia la extensión de lo ya estudiado y de estas nuevas lineas a problemas M-arios; que incluso podría albergar la posibilidad de obviar las limitaciones naturales de ciertas formulaciones de clasificación.

[1] Tal vez planteamientos similares sean incorporables a diseños bloque, utilizando una medida auxiliar para seleccionar los parámetros de la ponderación o de la forma de los ESTs.

Apéndice A

El algoritmo BP

Sea $\{(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})\}_{k=1}^{K}$ un conjunto de datos (etiquetados) de entrenamiento. El algoritmo BP para un MLP de L capas (cada capa tiene N_l neuronas; $l = 1, \ldots, L$) es el siguiente:

1. **Inicialización**: los pesos se inicializan de manera aleatoria.

2. **Presentación del conjunto de entrenamiento**: de forma cíclica (en "épocas") con las muestras en orden aleatorio.

3. **Propagación**: en el paso enésimo, la salida de la neurona j en la capa l es

$$o_j^{(l)}(n) = f_j^{(l)}\{\sum_{i=0}^{N_l} w_{ji}^{(l)}(n) o_i^{(l-1)}(n)\}, \quad l = 1, \ldots, L \tag{A.1}$$

   donde $N_0 = D$, N_l y $f_j^{(l)}$ son la dimensión de una muestra $\mathbf{x}$ del conjunto de entrenamiento, el número de neuronas ocultas de la capa l, y la activación aplicada a la salida de la neurona j de la capa l (se supone derivable), respectivamente; $o_i^{(l-1)}$ es la señal salida de la neurona i en la capa previa $l-1$, y $w_{ji}^{(l)}(n)$ es el peso de la conexión entre la neurona i de la capa $l-1$ y la neurona j de la capa l.

   Para $i = 0$, $o_0^{(l-1)}(n) = 1$ y $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$ que es el sesgo aplicado a la neurona j en la capa l. En la capa de entrada, $o_j^{(0)}(n) = [1 \ \mathbf{x}(n)]^{\mathrm{T}}$, siendo $\mathbf{x}(n)$ un vector de entrada al instante n.

El error a la salida es

$$e_j(n) = t_j(n) - o_j^{(L)}(n) \tag{A.2}$$

donde $t_j(n)$ es la j-ésima salida deseada del vector $\mathbf{t}(n)$. Típicamente, la función de coste es el error cuadrático medio $\frac{1}{2}e_j^2(n)$, y el error total de la capa de salida es

$$C(n) = \frac{1}{2}\sum_{j=1}^{N_L} e_j^2(n) \tag{A.3}$$

4. **Retropropagación**: el algoritmo BP aplica una corrección $-\eta^{(l)}\dfrac{\partial C(n)}{\partial w_{ji}^{(l)}(n)}$ a los pesos $w_{ji}^{(l)}(n)$ según la siguiente expresión

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) - \eta^{(l)}\,\frac{\partial C(n)}{\partial w_{ji}^{(l)}(n)} \tag{A.4}$$

siendo $\eta^{(l)}$ la tasa de aprendizaje correspondiente a la capa l. El cálculo de $\dfrac{\partial C(n)}{\partial w_{ji}^{(l)}(n)}$ se basa en la utilización de la regla de la cadena:

$$\frac{\partial C(n)}{\partial w_{ji}^{(l)}(n)} = \frac{\partial C(n)}{\partial o_j^{(l)}(n)}\,\frac{\partial o_j^{(l)}(n)}{\partial w_{ji}^{(l)}(n)} = \frac{\partial C(n)}{\partial o_j^{(l)}(n)}\,o_i^{(l-1)}(n) f_j^{'(l)} \tag{A.5}$$

siendo $f_j^{'(l)}$ la derivada de la activación $f_j^{(l)}$ de la neurona j de la capa l. A partir de (A.1) y (A.2), se calcula la derivada parcial de $C(n)$ con respecto a $o_j^{(l)}(n)$ y se deduce

$$\frac{\partial C(n)}{\partial o_j^{(l)}(n)} = \begin{cases} -e_j(n), & \text{para} \quad l = L \\ \displaystyle\sum_{k=1}^{N_{l+1}} \frac{\partial C(n)}{\partial o_k^{(l+1)}(n)} w_{kj}^{(l+1)}(n) f_k^{'(l+1)}, & \text{para} \quad l < L \end{cases} \tag{A.6}$$

5. **Iteración**: se iteran las etapas 3 y 4 hasta alcanzar un criterio de parada.

Apéndice B

Tablas de sensibilidad

En este apéndice presentamos las tablas de sensibilidad de los diseños EST-GP_{MLP}, EST-GP_{GPC}, y EST-GP_{SVM} para todos los problemas considerados en la parte experimental del capítulo 5 (los valores marcados en negrita corresponden a los valores encontrados por CV).

B.1. EST-GP_{MLP}

N	4	6	8	10	12	14	16
Contraceptive	71.89	71.83	71.88	71.87	**71.90**	71.85	72.01
Crabs	98.23	98.30	98.32	98.33	**98.35**	98.31	98.37
Credit	88.80	**88.80**	88.72	88.72	88.66	88.68	88.70
Hepatitis	88.36	87.83	88.04	87.81	87.94	**88.54**	88.04
Image	93.53	93.60	**93.62**	93.50	93.53	93.53	93.53
Ionosfera	95.54	95.71	95.65	95.58	**95.80**	95.73	95.55
Pima	78.37	78.47	78.30	78.58	**78.63**	78.56	78.78
Ripley	90.76	90.75	90.73	**90.78**	90.67	90.63	90.70

Tabla B.1: Sensibilidad de EST-GP_{MLP} con respecto a N.

μ	10^{-2}	0.1	0.3	0.6	1	1.2	1.6	2
Contraceptive	71.51	71.62	71.78	**71.90**	71.31	70.91	70.49	71.05
Crabs	96.37	96.36	96.84	**98.35**	97.18	96.96	96.96	96.96
Credit	88.74	88.79	**88.80**	88.56	88.53	88.82	89.66	90.12
Hepatitis	87.77	88.05	87.87	87.34	**88.54**	87.89	87.90	87.83
Image	88.74	88.80	88.80	89.11	91.05	92.03	**93.62**	93.10
Ionosfera	95.97	**95.80**	94.71	94.16	94.27	94.36	93.84	92.65
Pima	56.20	72.83	78.26	78.37	78.43	78.02	**78.63**	66.22
Ripley	90.56	90.61	90.70	**90.78**	90.44	90.45	90.54	90.52

Tabla B.2: Sensibilidad de EST-GP$_{\text{MLP}}$ con respecto a μ.

α_1	10^{-3}	10^{-2}	5.10^{-2}	0.1	0.5	1	1.5	2	3	4	5
Contraceptive	71.56	71.49	71.55	71.74	71.88	71.85	71.85	**71.90**	71.90	71.90	71.87
Crabs	98.60	**98.35**	97.49	97.02	97.17	96.88	96.88	96.88	96.88	96.78	96.88
Credit	88.70	88.70	88.75	88.75	88.78	88.77	88.79	88.78	88.79	**88.80**	88.66
Hepatitis	88.21	**88.54**	88.36	88.19	87.73	87.36	87.12	87.18	87.05	87.02	87.11
Image	89.45	89.83	90.60	91.28	93.13	**93.62**	93.46	93.39	93.35	93.24	93.22
Ionosfera	95.78	95.71	95.71	95.68	95.76	95.72	95.75	95.76	95.76	**95.80**	95.73
Pima	78.56	**78.63**	78.41	77.80	77.15	76.96	76.88	76.90	76.91	76.80	76.82
Ripley	90.54	90.60	90.67	90.75	**90.78**	90.76	90.76	90.76	90.77	90.77	90.77

Tabla B.3: Sensibilidad de EST-GP$_{\text{MLP}}$ con respecto a α_1.

α_2	10^{-3}	10^{-2}	5.10^{-2}	0.1	0.5	1	1.5	2	3	4	5
Contraceptive	70.31	70.27	70.36	70.44	71.23	71.80	71.82	**71.90**	71.68	71.58	71.54
Crabs	96.93	97.02	97.51	97.91	98.15	98.31	98.34	98.34	98.34	**98.35**	98.34
Credit	88.51	88.56	88.60	88.61	88.55	88.60	88.65	88.75	**88.80**	88.75	88.66
Hepatitis	88.48	**88.54**	88.36	87.75	86.75	85.97	85.32	85.11	84.47	83.86	83.90
Image	93.35	**93.62**	93.54	93.09	92.64	92.61	92.73	92.62	92.70	92.62	92.67
Ionosfera	95.94	**95.80**	95.21	94.99	94.82	94.53	94.53	94.40	94.30	93.93	93.72
Pima	78.51	**78.63**	78.64	78.70	78.46	78.36	78.19	78.21	78.17	78.23	78.20
Ripley	90.18	90.19	90.39	90.43	90.66	**90.78**	90.69	90.56	90.48	90.43	90.43

Tabla B.4: Sensibilidad de EST-GP$_{\text{MLP}}$ con respecto a α_2.

B.2. EST-GP$_{\text{GPC}}$

μ	10^{-2}	0.1	0.3	0.6	1	1.2	1.6	2
Contraceptive	64.25	65.53	67.42	69.15	70.53	**71.36**	70.32	70.44
Crabs	96.75	**97.00**	96.63	96.75	96.50	96.88	51.25	51.25
Credit	89.06	89.31	89.09	89.42	**90.94**	90.29	84.42	81.30
Hepatitis	87.10	89.35	**91.13**	85.65	83.23	85.49	85.00	92.58
Image	89.91	**89.50**	88.55	88.59	85.21	85.84	85.34	85.43
Ionosfera	92.33	92.60	92.47	92.07	**92.73**	92.80	91.73	91.73
Pima	76.10	76.00	75.67	75.97	76.19	76.19	**76.91**	76.52
Ripley	90.34	90.46	**90.67**	90.60	90.49	90.46	90.46	90.39

Tabla B.5: Sensibilidad de EST-GP$_{\text{GPC}}$ con respecto a μ.

α_1	10^{-3}	10^{-2}	5.10^{-2}	0.1	0.5	1	1.5	2	3	4	5
Contraceptive	67.41	66.53	66.68	68.78	70.78	71.27	71.24	**71.36**	71.41	71.15	70.97
Crabs	97.00	97.00	**97.00**	97.00	97.00	97.00	97.00	97.00	97.00	97.00	97.00
Credit	88.30	90.33	91.05	**90.94**	89.53	89.42	89.20	89.17	89.17	89.28	88.99
Hepatitis	91.13	91.13	91.13	91.13	91.13	91.13	**91.13**	91.13	91.13	91.13	91.13
Image	89.46	89.50	**89.50**	89.50	89.50	89.46	89.46	89.46	89.29	89.29	89.50
Ionosfera	87.07	90.60	91.87	92.47	92.40	92.53	**92.73**	92.60	92.47	92.33	92.27
Pima	73.88	73.00	75.31	76.16	76.42	76.68	76.68	76.74	**76.91**	76.74	76.65
Ripley	90.60	90.55	90.52	90.56	90.61	90.68	90.65	90.65	90.66	**90.67**	90.66

Tabla B.6: Sensibilidad de EST-GP$_{\text{GPC}}$ con respecto a α_1.

α_2	10^{-3}	10^{-2}	5.10^{-2}	0.1	0.5	1	1.5	2	3	4	5
Contraceptive	64.54	65.09	66.05	67.58	70.29	**71.36**	70.95	71.10	71.22	70.78	70.53
Crabs	51.25	51.25	51.25	51.25	96.13	96.63	96.63	**97.00**	96.50	96.50	96.50
Credit	89.06	89.02	89.24	89.35	89.60	90.22	90.36	90.76	90.98	**90.94**	91.12
Hepatitis	84.68	84.68	80.32	82.42	**91.13**	85.00	86.13	87.42	85.32	86.29	85.16
Image	85.33	85.74	87.29	85.63	85.52	87.42	88.88	88.18	**89.50**	88.90	88.55
Ionosfera	92.73	92.73	92.73	92.73	92.73	92.73	92.73	**92.73**	92.73	92.73	92.73
Pima	75.54	75.64	76.97	76.38	76.65	76.97	**76.91**	76.84	76.65	76.45	76.52
Ripley	49.68	54.21	67.21	78.24	89.83	**90.67**	90.63	90.71	90.53	90.53	90.49

Tabla B.7: Sensibilidad de EST-GP$_{\text{GPC}}$ con respecto a α_2.

B.3. EST-GP$_{\text{SVM}}$

C	0.1	1	10	100	10^3	10^4
Contraceptive	57.30	64.20	67.90	70.30	**71.20**	71.10
Crabs	96.90	97.40	**99.40**	98.60	98.30	98.30
Credit	49.3	80.50	**90.00**	87.80	87.00	88.00
Hepatitis	85.50	85.50	87.60	**90.30**	90.00	91.90
Image	86.10	91.80	93.70	93.80	**94.50**	92.50
Ionosfera	67.70	**96.90**	96.40	94.60	94.60	94.50
Pima	66.30	**79.30**	76.40	72.10	71.80	69.50
Ripley	90.50	**90.50**	90.50	90.30	90.40	90.40

Tabla B.8: Sensibilidad de EST-GP$_{\text{SVM}}$ con respecto a C.

σ	$2^{-5}\sqrt{D}$	$2^{-4}\sqrt{D}$	$2^{-3}\sqrt{D}$	$2^{-2}\sqrt{D}$	$2^{-1}\sqrt{D}$	$\sqrt{D}$	$2\sqrt{D}$	$2^{2}\sqrt{D}$	$2^{3}\sqrt{D}$	$2^{4}\sqrt{D}$	$2^{5}\sqrt{D}$
Contraceptive	70.70	69.90	69.50	67.70	69.20	71.10	**71.20**	70.40	68.50	68.50	67.20
Crabs	96.60	97.10	98.40	**99.40**	98.40	96.90	96.50	96.90	96.80	95.80	95.50
Credit	87.70	88.00	88.10	88.20	88.10	87.60	87.20	89.50	**90.00**	87.10	71.00
Hepatitis	83.90	83.90	84.50	88.10	90.80	91.30	**90.30**	83.70	87.10	85.70	85.50
Image	94.40	94.50	93.90	93.00	**94.50**	93.40	92.70	90.70	86.70	86.20	82.20
Ionosfera	92.40	91.80	92.70	**96.90**	96.70	95.80	72.70	68.90	56.50	56.50	56.50
Pima	76.50	76.30	75.90	78.00	**79.30**	77.50	67.40	77.20	65.30	65.20	65.20
Ripley	90.30	90.70	90.80	90.70	90.30	90.30	90.10	**90.50**	90.50	90.40	90.30

Tabla B.9: Sensibilidad de EST-GP_{SVM} con respecto a σ. D es la dimensión del dato de entrada.

μ	10^{-2}	0.1	0.3	0.6	1	1.2	1.6	2
Contraceptive	70.90	71.00	**71.20**	71.00	71.10	71.40	71.40	71.40
Crabs	95.60	95.60	95.50	96.30	99.00	**99.40**	99.40	97.00
Credit	93.60	93.30	92.00	**90.00**	88.40	87.40	85.90	86.60
Hepatitis	90.20	90.50	90.50	86.50	87.10	**90.30**	90.00	90.50
Image	94.50	94.50	**94.50**	93.00	94.40	94.30	94.50	94.50
Ionosfera	97.50	**96.90**	96.20	95.10	94.90	94.90	95.10	92.30
Pima	78.10	78.10	78.30	78.70	75.30	75.00	**79.30**	78.40
Ripley	88.90	89.10	90.10	89.90	90.40	90.50	**90.50**	87.50

Tabla B.10: Sensibilidad de EST-GP$_{\text{SVM}}$ con respecto a μ.

α_1	10^{-3}	10^{-2}	5.10^{-2}	0.1	0.5	1	1.5	2	3	4	5
Contraceptive	71.10	71.20	71.20	71.20	71.20	71.20	71.20	71.20	71.20	**71.20**	71.20
Crabs	96.00	96.00	98.00	99.50	**99.40**	99.00	98.90	98.00	97.80	97.30	97.40
Credit	89.50	89.80	89.90	89.90	90.00	90.00	**90.00**	90.00	90.00	90.00	90.00
Hepatitis	88.00	90.10	89.20	**90.30**	84.70	85.50	87.30	85.30	85.20	84.84	83.4
Image	94.50	94.50	**94.50**	94.40	94.40	94.40	94.40	94.40	94.40	94.40	94.40
Ionosfera	96.30	96.70	96.80	**96.90**	96.90	96.90	96.90	96.90	96.90	96.90	96.90
Pima	78.80	78.70	**79.30**	77.90	76.40	76.40	76.50	76.30	76.40	76.40	76.60
Ripley	64.60	64.70	68.10	83.60	**90.50**	90.60	90.60	90.80	90.70	90.70	90.50

Tabla B.11: Sensibilidad de EST-GP$_{\text{SVM}}$ con respecto a α_1.

α_2	10^{-3}	10^{-2}	5.10^{-2}	0.1	0.5	1	1.5	2	3	4	5
Contraceptive	71.00	71.10	**71.20**	71.20	71.10	71.20	71.40	71.40	71.40	71.30	71.30
Crabs	99.40	99.40	99.40	99.40	99.40	99.40	**99.40**	99.40	99.40	99.40	99.40
Credit	96.00	96.00	95.90	95.90	94.40	92.30	90.90	**90.00**	89.30	89.20	89.20
Hepatitis	88.60	87.90	88.60	**90.30**	87.70	87.60	86.50	86.50	86.50	86.60	86.30
Image	94.40	**94.50**	94.10	94.00	94.30	94.20	94.20	94.20	94.20	94.20	94.20
Ionosfera	97.30	**96.90**	96.50	96.40	94.90	94.70	94.50	94.70	94.30	93.90	93.50
Pima	78.90	78.90	79.30	**79.30**	79.30	79.30	79.20	79.20	79.20	79.20	79.20
Ripley	90.50	90.50	90.50	90.50	90.50	90.50	90.50	90.50	90.50	**90.50**	90.6

Tabla B.12: Sensibilidad de EST-GP$_{\text{SVM}}$ con respecto a α_2.

Apéndice C

El algoritmo EM

El algoritmo EM es un método iterativo para estimar los parámetros de la verosimilitud dado un conjunto de datos (suponiendo que los datos son independientes e idénticamente distribuidos)

A continuación, se presenta el algoritmo EM básico, y luego se describe su aplicación EM a modelos de mezcla de gaussianas.

C.1. El algoritmo EM básico

El algoritmo EM maximiza la verosimilitud $p(\mathbf{X}|\boldsymbol{\theta})$ dado los datos $\mathbf{X}$ buscando los parámetros $\boldsymbol{\theta}$ alternando dos pasos, expectación y maximización, del siguiente modo:

1. **Inicialización**: se inicializan los valores de los parámetros con $\boldsymbol{\theta}^{(0)}$.
2. **Etapa de Expectación** (**"E-step"**): se busca el valor medio de $\log p(\mathbf{X}|\boldsymbol{\theta})$ dada las observaciones $\mathbf{X}$ y la estimación actual $\boldsymbol{\theta}^{(i)}$ como

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = E[\log p(\mathbf{X}|\boldsymbol{\theta})|\mathbf{X}, \boldsymbol{\theta}^{(i)}] \tag{C.1}$$

3. **Etapa de Maximización** (**"M-step"**): se estiman los parámetros $\boldsymbol{\theta}^{(i+1)}$ maximizando la expectación $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ con respecto a $\boldsymbol{\theta}$

$$\boldsymbol{\theta}^{(i+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) \tag{C.2}$$

4. **Convergencia:** si $||\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^{(i)}|| < \epsilon$ ($\epsilon << 1$) se para el algoritmo; sino ir al paso 2.

5. Finalmente, devolver el valor estimado $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(i+1)}$.

C.2. Aplicación de EM a modelos de mezcla de gaussianas

Considerando un modelo de mezcla de L gaussianas

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{l=1}^{L} \pi_l p(\mathbf{X}|\mathbf{m}_l, \boldsymbol{\Sigma}_l) \tag{C.3}$$

con $\boldsymbol{\theta} = \{\pi_l, \mathbf{m}_l, \boldsymbol{\Sigma}_l\}_{l=1}^{L}$; $\{\pi_l\}$ $(0 \leq \pi_l \leq 1)$, $\{\mathbf{m}_l\}$ y $\{\boldsymbol{\Sigma}_l\}$ son los parámetros factor de mezcla (que cumplen $\sum_{l=1}^{L} \pi_l = 1$), los vectores medias y las matrices de covarianza. $p(\mathbf{X}|\mathbf{m}_l, \boldsymbol{\Sigma}_l)$ tiene la siguiente forma:

$$\begin{aligned} p(\mathbf{X}|\mathbf{m}_l, \boldsymbol{\Sigma}_l) &= \prod_{k=1}^{K} p(\mathbf{x}^{(k)}|\mathbf{m}_l, \boldsymbol{\Sigma}_l) \\ &= \prod_{k=1}^{K} \{ \frac{1}{(2\pi)^{D/2} \det(\boldsymbol{\Sigma}_l)^{1/2}} \exp[-\frac{1}{2}(\mathbf{x}^{(k)} - \mathbf{m}_l)^T \boldsymbol{\Sigma}_l^{-1} (\mathbf{x}^{(k)} - \mathbf{m}_l)] \} \\ &= \frac{1}{(2\pi)^{D\,K/2} \det(\boldsymbol{\Sigma}_l)^{K/2}} \exp[-\frac{1}{2} \sum_{k=1}^{K} (\mathbf{x}^{(k)} - \mathbf{m}_l)^T \boldsymbol{\Sigma}_l^{-1} (\mathbf{x}^{(k)} - \mathbf{m}_l)] \end{aligned} \tag{C.4}$$

La fórmula (C.3) resulta

$$p(\mathbf{X}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{D\,K/2}} \sum_{l=1}^{L} \frac{\pi_l}{\det(\boldsymbol{\Sigma}_l)^{K/2}} \exp[-\frac{1}{2} \sum_{k=1}^{K} (\mathbf{x}^{(k)} - \mathbf{m}_l)^T \boldsymbol{\Sigma}_l^{-1} (\mathbf{x}^{(k)} - \mathbf{m}_l)] \tag{C.5}$$

El algoritmo EM para estimar los parámetros $\boldsymbol{\theta}$ de la verosimilitud del modelo de mezcla de gaussianas es como sigue

1. **Inicialización**: $\boldsymbol{\theta}^0 = \{\pi_l^0, \mathbf{m}_l^0, \boldsymbol{\Sigma}_l^0\}_{l=1}^L$.

2. **E-step**: se calcula la expectación z_{kl}^i en la iteración i como

$$z_{kl}^i = \frac{\pi_l^i p(\mathbf{x}^{(k)}|\mathbf{m}_l^i, \boldsymbol{\Sigma}_l^i)}{\sum_{l'=1}^{L} \pi_{l'}^i p(\mathbf{x}^{(k)}|\mathbf{m}_{l'}^i, \boldsymbol{\Sigma}_{l'}^i)} \quad \text{para} \quad k = 1, \ldots, K, \quad l = 1, \ldots, L. \tag{C.6}$$

3. **M-step**: se maximiza la expectación de todo el conjunto de datos con respecto a cada parámetro de $\boldsymbol{\theta}$, obteniendo

$$\begin{aligned} \pi_l^{(i+1)} &= \frac{1}{K} \sum_{k=1}^{K} z_{kl}^i \\ \mathbf{m}_l^{(i+1)} &= \frac{1}{K} \sum_{k=1}^{K} z_{kl}^i \, \mathbf{x}^{(k)} \\ \boldsymbol{\Sigma}_l^{(i+1)} &= \frac{1}{K} \sum_{k=1}^{K} z_{kl}^i \, (\mathbf{x}^{(k)} - \mathbf{m}_l^{(i+1)})(\mathbf{x}^{(k)} - \mathbf{m}_l^{(i+1)})^T \end{aligned} \tag{C.7}$$

4. **Convergencia:** se itera hasta que el algoritmo converge o alcanzar un criterio de parada.

5. Finalmente, se devuelven los valores estimados de los parámetros $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(i+1)} = \{\pi_l^{(i+1)}, \mathbf{m}_l^{(i+1)}, \boldsymbol{\Sigma}_l^{(i+1)}\}_{l=1}^L$.

Apéndice D

Las aproximaciones de Laplace, EP, y EM-EP

D.1. Cálculo matricial de una distribución marginal y condicional gaussiana

Sea $\mathbf{x}$ y $\mathbf{y}$ dos vectores aleatorios que siguen una distribución gaussiana conjunta

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{m_x} \\ \mathbf{m_y} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{B} \end{bmatrix}) \tag{D.1}$$

La distribución marginal de $\mathbf{y}$ y condicional de $\mathbf{y}$ dada $\mathbf{x}$ son respectivamente,

$$\mathbf{y} \sim \mathcal{N}(\mathbf{m_y}, \mathbf{B}) \quad \text{y} \quad \mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{m_y} + \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}(\mathbf{x} - \mathbf{m_x}), \mathbf{B} - \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{C}) \tag{D.2}$$

D.2. Aproximación de Laplace

La aproximación de Laplace [Williams1998] reemplaza $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ por una densidad gaussiana, $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$

$$q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\mathbf{f}|\tilde{\mathbf{f}}, \mathbf{H}^{-1}) \tag{D.3}$$

donde $\tilde{\mathbf{f}} = \{\tilde{f}^{(k)}\}_{k=1}^{K}$ es la moda de $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$

$$\tilde{\mathbf{f}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) \tag{D.4}$$

y $\mathbf{H}$ es el Hessiano de $-\log p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ en $\tilde{\mathbf{f}}$. Así, la ecuación (5.8) se transforma en una convolución de dos gaussianas. Concretamente, la moda $\tilde{\mathbf{f}}$ se calcula utilizando el método iterativo de Newton

$$\mathbf{f} \longleftarrow \mathbf{f} - (\nabla\nabla_{\mathbf{f}} \log p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n))^{-1} \nabla_{\mathbf{f}} \log p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) \tag{D.5}$$

hasta la convergencia de $\mathbf{f}$. Entonces,

$$\tilde{\mathbf{f}} \longleftarrow \mathbf{f}; \quad \text{y} \quad \mathbf{H} = (\mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} + \mathbf{W}) \tag{D.6}$$

donde $\mathbf{W} = -\nabla\nabla_{\tilde{\mathbf{f}}} \log p(\mathbf{t}|\mathbf{f})$ es una matriz diagonal, y $\mathbf{C}_{\boldsymbol{\theta},\theta_n}$ es la matriz de covarianza de $p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$, con hiperparámetros $\boldsymbol{\theta}$ y θ_n.

El cálculo de $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ facilita la aproximación de la verosimilitud marginal $q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$ que tiene la siguiente forma

$$\log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) = -\frac{1}{2}\tilde{\mathbf{f}}^T \mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} \tilde{\mathbf{f}} + \log p(\mathbf{t}|\tilde{\mathbf{f}}) - \frac{1}{2}\log|\mathbf{B}| \tag{D.7}$$

siendo $\mathbf{B} = \mathbf{I} + \mathbf{W}^{1/2} \mathbf{C}_{\boldsymbol{\theta},\theta_n} \mathbf{W}^{1/2}$ ($\mathbf{I}$ es la matriz identidad de dimensión $K \times K$).

Seguidamente, se maximiza la verosimilitud marginal aproximada $q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$ con respecto a θ (uno de los hiperparámetros). Evidentemente, $\mathbf{C}_{\boldsymbol{\theta},\theta_n}$ es función de $\boldsymbol{\theta}$ y θ_n, y también $\tilde{\mathbf{f}}$ y $\mathbf{W}$ lo son implícitamente. Si $\boldsymbol{\theta}$ o θ_n varían, la moda $\tilde{\mathbf{f}}$ cambia.

La derivada parcial de $\log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$ con respecto a θ se descompone en dos términos

$$\frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial\theta} = \frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial\theta}\Big|_{\text{explícita}} + \frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial\theta}\Big|_{\text{implícita}} \tag{D.8}$$

donde se calcula la derivada parcial explícita como sigue:

$$\frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial\theta}\Big|_{\text{explícita}} = \frac{1}{2}\tilde{\mathbf{f}}^{\mathrm{T}} \mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta} \mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} \tilde{\mathbf{f}} - \frac{1}{2} tr((\mathbf{C}_{\boldsymbol{\theta},\theta_n} + \mathbf{W}^{-1})^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta}) \tag{D.9}$$

y la derivada parcial implícita tiene la siguiente forma:

$$\frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial \theta}|_{\text{implícita}} = \sum_{k=1}^{K} \frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial \tilde{f}^{(k)}} \frac{\partial \tilde{f}^{(k)}}{\partial \theta} \tag{D.10}$$

donde

$$\frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial \tilde{f}^{(k)}} = -\frac{1}{2} tr(\mathbf{B}^{-1} \mathbf{C}_{\boldsymbol{\theta}, \theta_n} \frac{\partial \mathbf{W}}{\partial \tilde{f}^{(k)}}) \tag{D.11}$$

y

$$\frac{\partial \tilde{\mathbf{f}}}{\partial \theta} = \mathbf{B}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\theta}, \theta_n}}{\partial \theta} \nabla \log p(\mathbf{t}|\tilde{\mathbf{f}}) \tag{D.12}$$

La complejidad computacional para la implementación del método de Laplace está dominada por la descomposición de Cholesky para el cálculo de $\mathbf{B}$, que requiere $K^3/6$ operaciones (se refiere al número de iteraciones de Newton), y el resto de las operaciones del algoritmo de Laplace son de orden $O(K)$.

En cuanto a las limitaciones del método de Laplace, cabe destacar el desajuste de la densidad de probabilidad predictiva sobre f^*, $p(f^*|\mathbf{x}^*, \mathbf{X}, \mathbf{t}, \boldsymbol{\theta}, \theta_n)$, para problemas de alta dimensión [Kuss2005].

D.3. Aproximación EP

El algoritmo EP se basa en la inferencia Bayesiana [Minka2001a] para aproximar la distribución $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ por una gaussiana $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ de vector media $\mathbf{m}$ y matriz de covarianza $\boldsymbol{\Sigma}$.

Concretamente, se aproxima la verosimilitud (no gaussiana) $p(t^{(k)}|f^{(k)})$ con una aproximación local $\tilde{g}_k(f^{(k)})$,

$$\tilde{g}_k(f^{(k)}) = s^{(k)} \exp(-\frac{(f^{(k)} - \mu^{(k)})^2}{2v^{(k)}}), \quad \text{para} \quad k = 1, \ldots, K \tag{D.13}$$

$\mu^{(k)}$, $v^{(k)}$, y $s^{(k)}$ son términos que se ajustan con el algoritmo EP.

Por tanto, $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ se escribe como

$$q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) \propto \prod_{k=1}^{K} \tilde{g}_k(f^{(k)})\ p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) = \tilde{g}_0(\mathbf{f}) \prod_{k=1}^{K} \tilde{g}_k(f^{(k)}) \tag{D.14}$$

siendo $\tilde{g}_0(\mathbf{f}) = p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$.

A partir de (D.13) y de $p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(\mathbf{0}, \mathbf{C}_{\boldsymbol{\theta},\theta_n})$, $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ en la ecuación (D.14) se convierte en un producto de términos sencillos que pertenecen a la familia exponencial. Las expresiones de $\mathbf{m}$ y $\boldsymbol{\Sigma}$ se deducen a través de

$$\mathbf{m} = \boldsymbol{\Sigma}\mathbf{V}^{-1}\boldsymbol{\mu} \quad \text{y} \quad \boldsymbol{\Sigma} = (\mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} + \mathbf{V}^{-1})^{-1} \tag{D.15}$$

donde $\mathbf{V} = \text{diag}(v^{(1)}, \ldots, v^{(K)})$ es una matriz diagonal y $\boldsymbol{\mu} = [\mu^{(1)}, \ldots, \mu^{(K)}]^T$.

A continuación, se presenta el procedimiento EP para ajustar iterativamente los $(\mu^{(k)}, v^{(k)}, s^{(k)})$.

Se inicializan los valores $\mu^{(k)}$, $v^{(k)}$, $s^{(k)}$, m_k, y σ_k del siguiente modo:

$\mu^{(k)} = 0, \quad v^{(k)} = \infty, \quad s^{(k)} = 1, \quad m_k = 0, \quad \text{y} \quad \sigma_k = c_{\boldsymbol{\theta},\theta_n}(\mathbf{x}^{(k)}, \mathbf{x}^{(k)})$

donde $m_k = E[f^{(k)}]$ y $\sigma_k = Var[f^{(k)}]$ son la media y la varianza de $q(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$, respectivamente.

Para $k = 1, \ldots, K$:

1. Eliminar el k-ésimo término $\tilde{g}_k(f^{(k)})$ de $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ para producir una aproximación "antigua" $q^{\setminus k}(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ y deducir

$$q^{\setminus k}(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(m_k^{\setminus k}, \sigma_k^{\setminus k}) \tag{D.16a}$$

donde

$$m_k^{\setminus k} = E[f^{(k)\,\setminus k}] = m_k + \sigma_k^{\setminus k}(v^{(k)})^{-1}(m_k - \mu^{(k)}) \tag{D.16b}$$

$$\sigma_k^{\setminus k} = Var[f^{(k)\,\setminus k}] = (\frac{1}{\sigma_k} - \frac{1}{v^{(k)}})^{-1} \tag{D.16c}$$

2. Buscar la nueva $q_{new}(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) = \mathcal{N}(m_k, \sigma_k)$ minimizando la divergencia de Kullback-Leibler de $q^{\setminus k}(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)p(t^{(k)}|f^{(k)})$ con respecto a $q(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$:

$q_{new}(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n) =$

$$\underset{q(f^{(k)}|\mathbf{t},\mathbf{X},\boldsymbol{\theta},\theta_n)}{\text{argmín}} \{\text{KL}(q^{\setminus k}(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)p(t^{(k)}|f^{(k)})||q(f^{(k)}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n))\} \tag{D.17}$$

con

$$m_k = m_k^{\setminus k} + \sigma_k^{\setminus k}\alpha_k \ ; \quad \alpha_k = \frac{t^{(k)}(1-Z_k)}{\sqrt{1+\sigma_k^{\setminus k}}} \ ; \quad Z_k = sgm(z_k) \ ; \quad z_k = \frac{t^{(k)}m_k^{\setminus k}}{\sqrt{1+\sigma_k^{\setminus k}}} \tag{D.18}$$

y

$$\sigma_k = \sigma_k^{\setminus k} + (\sigma_k^{\setminus k})^2\beta_k \ ; \quad \beta_k = -\frac{(t^{(k)})^2 Z_k(1-Z_k)}{1+\sigma_k^{\setminus k}} \tag{D.19}$$

3. Calcular la nueva $\tilde{g}_k(f^{(k)})$:

$$\begin{aligned} v^{(k)} &= (\sigma_k^{-1} - (\sigma_k^{\setminus k})^{-1})^{-1} \\ \mu^{(k)} &= v^{(k)}(\sigma_k^{-1}m_k - (\sigma_k^{\setminus k})^{-1}m_k^{\setminus k}) \\ s^{(k)} &= Z_k\sqrt{2\pi}\sqrt{\sigma_k + \sigma_k^{\setminus k}}\exp(\frac{1}{2}\frac{(m_k^{\setminus k} - m_k)^2}{\sigma_k + \sigma_k^{\setminus k}}) \end{aligned} \tag{D.20}$$

4. Obtener la nueva $q(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ mediante el cálculo de $\boldsymbol{\Sigma}$ y $\mathbf{m}$ con los nuevos valores de $\mu^{(k)}$ y $v^{(k)}$ de (D.20).

Finalmente, la aproximación de la verosimilitud marginal $q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$ se obtiene normalizando la ecuación (D.14):

$$\begin{aligned} &\log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) = \log \int p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) \prod_{k=1}^{K} \tilde{g}_k(f^{(k)}) \ d\mathbf{f} \\ &= \sum_{k=1}^{K} \log s^{(k)} - \frac{1}{2}\log|\mathbf{C}_{\boldsymbol{\theta},\theta_\mathbf{n}} + \mathbf{V}| - \frac{1}{2}\boldsymbol{\mu}^T(\mathbf{C}_{\boldsymbol{\theta},\theta_n} + \mathbf{V})^{-1}\boldsymbol{\mu} - \frac{K}{2}\log 2\pi \end{aligned} \tag{D.21}$$

La adaptación de los hiperparámetros se hace maximizando $\log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$ con respecto a θ

$$\begin{aligned} \frac{\partial \log q(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{\partial\theta} &= \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}(\mathbf{C}_{\boldsymbol{\theta},\theta_n} + \mathbf{V})^{-1}\frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta}(\mathbf{C}_{\boldsymbol{\theta},\theta_n} + \mathbf{V})^{-1}\boldsymbol{\mu} \\ &\quad -\frac{1}{2}tr((\mathbf{C}_{\boldsymbol{\theta},\theta_n} + \mathbf{V})^{-1}\frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial\theta}) \end{aligned} \tag{D.22}$$

Minka demostró que este algoritmo proporciona -en general- mejores prestaciones que el método de Laplace y el método variacional de Bayes. Se tiene una complejidad computacional de orden de $O(K^3)$ debido a la inversión de matrices; además, EP puede encontrarse con dificultades de convergencia, dependiendo del problema que se considere [Minka2001a].

D.4. Aproximación EM-EP

EM-EP [Kim2006] es una modificación del procedimiento EP que incluye la estimación de $\boldsymbol{\theta}$ y θ_n mediante la aplicación de EM en dos pasos:

- **Paso E**: se aproxima $p(\mathbf{f}|\mathbf{t}, \mathbf{X}, \boldsymbol{\theta}, \theta_n)$ por una gaussiana

$$q(\mathbf{f}) = \mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma}) \tag{D.23}$$

  aplicando EP. $\mathbf{m}$ y $\boldsymbol{\Sigma}$ son el vector media y la matriz de covarianza de $q(\mathbf{f})$, respectivamente.

- **Paso M**: dada $q(\mathbf{f})$, se reestiman $\boldsymbol{\theta}$ y θ_n maximizando el límite inferior $F(\boldsymbol{\theta}, \theta_n)$ de la desigualdad de Jensen

$$\begin{aligned} F(\boldsymbol{\theta}, \theta_n) &= \int q(\mathbf{f}) \log \frac{P(\mathbf{t}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)}{q(\mathbf{f})}\, d\mathbf{f} \\ &\leq \log P(\mathbf{t}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) \end{aligned} \tag{D.24}$$

Considerando la forma gaussiana de $p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)$, se obtienen los nuevos valores de $\boldsymbol{\theta}$ y θ_n derivando $F(\boldsymbol{\theta}, \theta_n)$ con respecto a θ:

$$\begin{aligned} \frac{\partial F(\boldsymbol{\theta}, \theta_n)}{\partial \theta} = &-\frac{1}{2}\, tr(\mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1}\, \frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial \theta}) + \frac{1}{2}\mathbf{m}^T\, \mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial \theta}\, \mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1}\mathbf{m} \\ &+\frac{1}{2}\, tr(\mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1}\, \frac{\partial \mathbf{C}_{\boldsymbol{\theta},\theta_n}}{\partial \theta}\mathbf{C}_{\boldsymbol{\theta},\theta_n}{}^{-1}\boldsymbol{\Sigma}) \end{aligned} \tag{D.25}$$

Los dos pasos se alternan hasta la convergencia. El detalle de la derivación de la ecuación (D.25) se encuentra más adelante.

El algoritmo EM-EP suele ofrecer buenas estimaciones de los hiperparámetros; su principal inconveniente es su alto coste computacional, ($O(K^3)$ operaciones).

Derivación de $F(\boldsymbol{\theta}, \theta_n)$ con respecto a θ

$F(\boldsymbol{\theta}, \theta_n)$ se puede escribir de la siguiente forma

$$F = \int q(\mathbf{f}) \log P(\mathbf{t}|\mathbf{f}) d\mathbf{f} + \int q(\mathbf{f}) \log p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) d\mathbf{f} - \int q(\mathbf{f}) \log q(\mathbf{f}) d\mathbf{f} \quad \text{(D.26)}$$

La primera y la última integral son independientes de $\boldsymbol{\theta}$ y θ_n; entonces

$$\frac{\partial F}{\partial \theta} = \frac{\partial (\int q(\mathbf{f}) \log p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) d\mathbf{f})}{\partial \theta} \quad \text{(D.27)}$$

y

$$\begin{aligned}
\int q(\mathbf{f}) \log p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n) d\mathbf{f} &= E_q[\log p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta}, \theta_n)] \\
&= E_q[-\tfrac{1}{2} \log |2\pi \mathbf{C}_{\boldsymbol{\theta}, \theta_n}| - \tfrac{1}{2} \mathbf{f}^T {\mathbf{C}_{\boldsymbol{\theta}, \theta_n}}^{-1} \mathbf{f}] \\
&= -\frac{1}{2} \log |2\pi \mathbf{C}_{\boldsymbol{\theta}, \theta_n}| - \frac{1}{2} E_q[\mathbf{f}^T {\mathbf{C}_{\boldsymbol{\theta}, \theta_n}}^{-1} \mathbf{f}] \\
&= -\frac{1}{2} \log |2\pi \mathbf{C}_{\boldsymbol{\theta}, \theta_n}| - \frac{1}{2} E_q[\mathbf{f}]^T {\mathbf{C}_{\boldsymbol{\theta}, \theta_n}}^{-1} E_q[\mathbf{f}] - \frac{1}{2} tr({\mathbf{C}_{\boldsymbol{\theta}, \theta_n}}^{-1} Cov[\mathbf{f}]) \\
&= -\frac{1}{2} \log |2\pi \mathbf{C}_{\boldsymbol{\theta}, \theta_n}| - \frac{1}{2} \mathbf{m}^T {\mathbf{C}_{\boldsymbol{\theta}, \theta_n}}^{-1} \mathbf{m} - \frac{1}{2} tr({\mathbf{C}_{\boldsymbol{\theta}, \theta_n}}^{-1} \boldsymbol{\Sigma})
\end{aligned} \quad \text{(D.28)}$$

Finalmente, se calculan las derivadas parciales de (D.28) con respecto a θ, y se obtiene la expresión (D.25).

Bibliografía

[Aha1991] D. W. Aha, "Incremental Constructive Induction: An Instance-Based Approach", In *Proceedings of the* 8^{th} *International Workshop on Machine Learning*, Evanston, MI: Morgan Kaufmann, pp. 117-121, 1991.

[Akaike1973] H. Akaike, "Information Theory and An Extensionof the Maximum Likelihood Principle", In B.Ñ. Petrov and F. Csáki (Eds.),2^{nd} *International Symposium on Information Theory*, pp. 267-281; Tsahkadsov, Armenia, USSR, 1973.

[Almeida2000] M. B. Almeida, A. Braga, J. P. Braga, "SVM-KM: Speeding SVMs Learning with a Priori Cluster Selection and k-Means", *Proceedings of the* 6^{th} *Brazilian Symposium on Neural Networks*, pp. 162-167; Rio de Janeiro, Brazil, 2000.

[Archambeau2003] C. Archambeau, J. A. Lee, M. Verleysen, "On Convergence Problems of the EM Algorithm for Finite Mixture Models", *Proceedings of the* 11^{th} *European Symposium on Artificial Neural Networks*, pp. 99-106; Bruges (Belgium), 2003.

[Archambeau2004] C. Archambeau, F. Vrins, M. Verleysen, "Flexible and Robust Bayesian Classification by Finite Mixture Models", *Proceedings of the* 12^{th} *European Symposium on Artificial Neural Networks*, pp. 75-80; Bruges (Belgium), 2004.

[Anderson1977] J. A. Anderson, J. W. Silverstein, S. A. Ritz, R. S. Jones, "Distinctive Features, Categorical Perception, and Probability Learning:

Some Applications of a Neural Model", *Psychological Review*, vol. 84, pp. 413-451, 1977.

[Bishop2006] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, NY: Springer, 2006.

[Blake] C. L. Blake, C. J. Merty, UCI Repository of Machine Learning Databases: www.ics.uci.edu/∼mlearn

[Boser1992] B. Boser, I. Guyon, V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", *Proceedings of the 5^{th} Annual Workshop on Computational Learning Theory*, pp. 144-152; Pittsburgh, PA, 1992.

[Breiman1984] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *Classification and Regression Trees*, Belmont, CA: Wadsworth International Group, 1984.

[Breiman1998] L. Breiman, *Arcing classifier*, *Annals of Statistics*, vol. 26(3), pp. 801-849, 1998.

[Buntine1994] W. L. Buntine, "A Guide to the Literature on Learning Probabilistic Networks from Data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 8(2), pp. 195-210, 1994.

[Burrascano1991] P. Burrascano, "A New Selection Criterion for the Generalized Delta Rule", *IEEE Transactions on Neural Networks*, vol. 2, pp. 125-130, 1991.

[Burges1998] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining Knowledge Discovery*, vol. 2(2), pp. 121-167, 1998.

[Cachin1994] C. Cachin, "Pedagogical Pattern Selection Strategies", *Neural Networks*, vol. 7, pp. 171-181, 1994.

[Chang1993] E. I. Chang, R. P. Lippmann, "A Boundary Hunting Radial Basis Function Classifier which Allocates Centers Constructively", in *Advances in Neural Information Processing Systems 5*, S. J. Hanson, J.

D. Cowan, C. L. Giles (eds.), pp. 139-146; San Mateo, CA: Morgan Kaufmann, 1993.

[Cheung1992] R. K. M. Cheung, I. Lusting, A. L. Kornhauser, "Relative Effectiveness of Training Set Patterns for Back Propagation", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, pp. 673-678; San Diego, CA, 1992.

[Cherkassky1998] V. Cherkassky, F. Mulier, "Learning from Data", A Wiley-Interscience Publication, John Wiley & Sons, Inc., 1998.

[Choi2002] S. H. Choi, P. Rockett, "The Training of Neural Classifiers with Condensed Datasets", *IEEE Transactions on Systems, Man, and Cybernetics*, Pt. B, vol. 32, pp. 202-206, 2002.

[Cichocki1993] A. Cichocki, R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, New York: Wiley, 1993.

[Cortes1995] C. Cortes, V. Vapnik, "Support Vector Networks", *Machine Learning*, vol. 20, pp. 273-297, 1995.

[Darken1992] C. Darken, J. Chang, J. Moody, "Learning Rate Schedules for Faster Stochastic Gradient Search", in *Neural Networks for Signal Processing II, pp. 3-22; New York. IEEE; 1992.*

[Dempster1977] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Series B (Methodological), vol. 39(1), pp. 1–38, 1977.

[Denker1987] J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, "Large Automatic Learning, Rule Extraction, and Generalization", *Complex Systems*, Complex Systems Publications, Inc., vol. 1, pp. 877-922, 1987.

[Duda2001] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification*, New York: John Wiley and Sons, 2^{nd} edition, 2001.

[El Jelali2008a] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "An Emphasized Target Smoothing Procedure to Improve MLP Classifiers Performance", *Proceedings of the* 16^{th} *European Symposium on Artificial Neural Networks*, pp. 499-504; Bruges, Belgium, 2008.

[El Jelali2008b] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "Applying Emphasized Soft Target for Gaussian Mixture Model Based Classification", *Proceedings of the International Multiconference on Computer Science and Information Technology*, 3^{rd} *International Symposium Advances in Artificial Intelligence and Applications*, vol. 3, pp. 131-136; Wisla, Poland, 2008.

[El Jelali2009] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "Designing Model Based Classifiers by Emphasizing Soft Targets", *Fundamenta Informaticae*, vol. 96(4), pp. 419-433, 2009.

[El Jelali2011] S. El Jelali, A. Lyhyaoui, A. R. Figueiras-Vidal, "Emphasized Soft Targets for Gaussian Process Classifier Design", submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[Elman1990] J. L., Elman, "Finding Structure in Time", *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[Falhman1988] S. E. Falhman, "Faster Learning Variations on Back-Propagation: An Emperical Study", in *Proc. 1988 Connectionist Models; Summer School, pp. 38-51; 1988.*

[Fisher1936] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems", *Annals of Eugenics*, vol. 7, Pt. II, pp. 179-188, 1936.

[Franco2000] L. Franco, S. A. Cannas, "Generalization and Selection of Examples in Feed-Forward Neural Networks", *Neural Computation*, vol. 12, pp. 2405-2426, 2000.

[Freund1996a] Y. Freund, R. E. Schapire, "Experiments with a New Boosting Algorithm", *Proceedings of the* 13^{th} *International Conference on Machine Learning*, pp. 148-156; Bari, Italy, 1996.

[Freund1996b] Y. Freund, R. E. Schapire, "Game Theory, on-line Prediction, and Boosting", *Proceedings of the 9^{th} Annual Conference on Computational Learning Theory*, pp. 325-332; Desenzano di Garda, Italy, 1996.

[Fukumizu1994] K. Fukumizu, S. Watanabe, "Error Estimation and Learning Data Arrangement for Neural Networks", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 777-780, 1994.

[Fukunaga1990] K. Fukunaga, *Introduction to Statistic Pattern Recognition*, New York: Academic Press, 2^{nd} edition, 1990.

[García-Pedrajas2009] N. García-Pedrajas, "Constructing Ensembles of Classifiers by Means of Weigthed Instance Selection", *IEEE Transactions on Neural Networks*, vol. 20(2), pp. 258-277, 2009.

[Gorse1997] D. Gorse, A. J. Shepperd, J. G. Taylor, "The New ERA in Supervised Learning", *Neural Networks*, vol. 10, pp. 343-352, 1997.

[Gómez-Verdejo2006] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, A. R. Figueiras-Vidal, "Boosting by Weighting Critical and Erroneous Samples", *Neurocomputing*, vol. 69, pp. 679-685, 2006.

[Gómez-Verdejo2008] V. Gómez-Verdejo, J. Arenas-García, A. R. Figueiras-Vidal, "A Dynamically Adjusted Mixed Emphasis Method for Building Boosting Ensembles", *IEEE Transactions on Neural Networks*, vol. 19, pp. 3-17, 2008.

[GPML] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning: www.GaussianProcess.org/gpml

[Grossberg1972] S. Grossberg, "Neural Expectation: Cerebellar and Retinal Analogs of Cells Fired by Learnable or Unlearned Pattern Classes", *Kybernetik*, vol. 10, pp. 49-57, 1972.

[Grossberg1976a] S. Grossberg, "Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Detectors", *Biological Cybernetics*, vol. 23, pp. 121-134, 1976.

[Grossberg1976b] S. Grossberg, "Adaptive Pattern Classification and Universal Recording: II. Feedback, Expectation, Olfaction, Illusions", *Biological Cybernetics*, vol. 23, pp. 187-202, 1976.

[Hart1968] P. E. Hart, "The Condensed Nearest Neighbor Rule", *IEEE Transactions on Information Theory*, vol. 14, pp. 515-516, 1968.

[Haykin1999] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, 2^{nd} edition, 1999.

[Hebb1949] D. O. Hebb, *The Organization of Behavior*, Wiley: New York; 1949.

[Hopfield1982] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the National Academy of Sciences of the USA*, vol. 79, pp. 2254-2558, 1982.

[Hopfield1987] J. J. Hopfield, "Learning Algorithm and Probability Distributions in Feed-Forward and Feed-Back Networks" *Proceeding of the National Academy of Sciences USA*, vol. 84, pp. 8429-8433, 1987.

[Huyser1988] K. Huyser, A. M. Horowitz, "Generalization in Connectionist Networks that Realize Boolean Functions" in D. Touretzky, G. Hinton, and T. Sejnowski. (eds.), *Proceedings of the Connectionist Models Summer School*, Morgan Kaufmann, pp. 191-200; Palo Alto, CA, 1988.

[Ishibuchi1994] H. Ishibuchi, A. Miyazaky, "Determination of Inspection Order for Classifying New Samples by Neural Networks", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 5, pp. 2907-2910, 1994.

[Jacobs1991] R. A. Jacobs, M. I. Jordan, "A Competitive Modular Connectionist Architecture", *Advances in Neural Information Processing Systyems*, vol. 5, (D. Touretzky, Ed.), Morgan Kaufmann, pp. 767-773; San Mateo, CA, 1991.

[Jiang1994] M. I. Jordan, R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm", *Neural Computation*, vol. 6, pp. 181-214, 1994.

[Kim2006] H.-C. Kim, Z. Ghahramani, "Bayesian Gaussian Process Classification with the EM-EP Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1948-1959, 2006.

[Kohonen1982] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps", *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

[Kohonen1984] T. Kohonen, Kohonen, T. "Self-Organization and Associative Memory", (3rd edition 1989), Springer, Berlin, 1984.

[Kung1995] S. Y. Kung, J. S. Taur, "Decision-Based Neural Networks with Signal/Image Classification Applications", *IEEE Transactions on Neural Networks*, vol. 6, pp. 170-181, 1995.

[Kuss2005] M. Kuss, C. E. Rasmussen, "Assessing Approximate Inference for Binary Gaussian Process Classification", *Journal of Machine Learning Research*, vol. 6, pp. 1679-1704, 2005.

[Kwok1999] J. T. Kwok, "Moderating the Output of Support Vector Classifiers", *IEEE Transactions on Neural Networks*, vol. 10, pp. 1018-1031, 1999.

[Lang1990] K. J. Lang, A. H. Waibel, G. E. Hinton, "A Time-Delay Neural Network Architecture for Isolated Word Recognition", *Neural Networks*, vol. 3, pp. 23-43, 1990.

[LeCun1985] Y. LeCun, "Une procedure dápprentissage pour reseau a seuil assymetrique", *Cognitiva*, vol. 85, pp. 599-604, 1985.

[Leisch1998] F. Leisch, L. C. Jain, K. Hornik, "Cross-Validation with Active Pattern Selection for Neural Network Classifiers", *IEEE Transactions on Neural Networks*, vol. 9, pp. 35-41, 1998.

[Lowe1989] D. Lowe, "Adaptive Radial Basis Function Nonlinearities, and the Problem of Generalization", *Proceedings 1st IEEE International Conference on Artificial Neural Networks*, pp. 171-175; London (UK), 1989.

[Lyhyaoui1999] A. Lyhyaoui, M. Martinez-Ramón, I. Mora-Jiménez, M. Vázquez-Castro, J. L. Sancho-Goméz, A. R. Figueiras-Vidal,

"Sample Selection via Clustering to Construct Support Vector-like Classifiers", *IEEE Transactions on Neural Networks*, vol. 10, pp. 1474-1481, 1999.

[MacKay1991] D. J. C. MacKay, *Bayesian Modeling and Neural Networks*, Ph. D. Thesis, Computation and Neural Systems, California Inst. Technology, Pasadena, CA, 1991.

[MacKay2003] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge: Cambridge University Press, 2003.

[Marchand1993] M. Marchand, M. Golea, "An Approximation Algorithm to Find the Largest Linearly Separable Subset of Training Examples", *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 556-559; Nagoya, Japan, 1993.

[Matheus1989] C. J. Matheus, L. A. Rendell, "Constructive induction on decision trees". In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*; Detroit, MI: Morgan Kaufmann, pp. 645-650, 1989.

[Matheus1990] C. J. Matheus, "Adding domain knowledge to SBL through feature construction", In *Proceedings of the Eighth National Conference on Artificial Intelligence*; Boston, MA: AAAI Press, pp. 803-808, 1990.

[McCulloch1943] W. S. McCulloch, W. Pitts, "A Logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.

[Michalsky1980] R. Michalsky, "Pattern Recognition as Rule-Guided Inductive Inference", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2(4), pp. 349-361, 1980.

[Minsky1969] M. L. Minsky, S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, 1969.

[Minka2001a] T. P. Minka, *A Family of Algorithms for Approximate Bayesian Inference*, PhD thesis. Massachusetts Institute of Technology, January, 2001.

[Minka2001b] T. P. Minka, *The EP Energy Function and Minimization Schemes*, Technical report TR2000-26, MIT Media Lab, 2001.

[Moody1992] J. E. Moody, "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems", *Advances in Neural Information Processing Systems*, (J. E. Moody, S. J. Hanson, and R. P. Lippmann, eds), pp. 847-854; San Mateo, CA: Morgan Kaufmann, 1992.

[Mora-Jiménez2009] I. Mora-Jiménez, A. R. Figueiras-Vidal, "Improving Performance of Neural Classifiers via Selective Reduction of Target Levels", *Neurocomputing*, vol. 72(13-15), pp. 3020-3027, 2009.

[Munro1992] P. W. Munro, "Repeat until Bored: A Pattern Selection Strategy", in J.E. Moody et al., eds., *Advances in Neural Information Processing System*, vol. 4, pp. 1001-1008; San Mateo, CA, Morgan Kaufmann, 1992.

[Müller2001] K. R. Müller,S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithm", *IEEE Transactions Neural Networks*, vol. 12, pp. 181-201, 2001.

[Nadaraya1964] É A. Nadaraya, "On Estimating Regression", *Theory of Probability and Its Applications*, vol. 9, pp. 141-412, 1964.

[Neal1993] R. M. Neal, *Probabilistic inference using Markov chain Monte Carlo methods*, Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[Niyogi1996] P. Niyogi, F. Girosi, "On the Relationship between Generalization Error, Hypothesis Complexity, and Sample Complexity for Radial Basis Functions", *Neural Computation*, vol. 8, pp. 819-842, 1996.

[Ohnishi1991] N. Ohnishi, A. Okamoto, N. Sugi, "Selective Presentation of Learning Samples for Efficient Learning in Multilayer Perceptron", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, pp. 688-690; Washington, D.C., 1991.

[Osuna1997] E. Osuna, R. Freund, F. Girosi, "An Improved Training Algorithm for Support Vector Machines", in *Proceedings Neural Networks for Signal Processing*, pp. 276-285; Amelia Island, FL, 1997.

[Parker1985] D. B. Parker, "Learning-Logic: Casting the Cortex of the Human Brain in Silicon", Tecnical Report TR-47. Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985.

[Parzen1962] E. Parzen, "On Estimation of a Probability Density Function and Mode", *Annals of Mathematical Statistics*, vol. 33, pp. 1065-1076, 1962.

[Pérez-Cruz2001] F. Pérez-Cruz, "IRWLS Matlab Toolbox to Solve the SVM for Pattern Recognition and Regression Estimation", 2002. Available: http://www.tsc.uc3m.es/∼fernando/

[Plutowski1993] M. Plutowski, H. White, "Selecting Concise Training Sets from Clean Data", *IEEE Transactions on Neural Networks*, vol. 4, pp. 305-318, 1993.

[PRNN] B. D. Ripley, Pattern Recognition and Neural Networks: http://www.stats.ox.ac.uk/pub/PRNN

[Quinlan1979] J. R. Quinlan, *Discovering Rules from Large Collections of Examples, en Expert Systems in the Microelectronic Age*, (D. Michie, Ed.), Edimburgo University Press, Edimburgo, 1979.

[Rasmussen2006] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, Cambridge, MA: The MIT Press, 2006.

[Reed1992] R. Reed, S. Oh, and R. J. Marks, II, "Regularization using Jittered Training Data", *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 147-152; Baltimore, MD, 1992.

[Reed1995] R. Reed, S. Oh, and R. J. Marks, II, "Similarities of Error Regularization, Sigmoid Gain Scaling, Target Smoothing, and Training with Jitter", *IEEE Transactions on Neural Networks*, vol. 6, pp. 529-538, 1995.

[Ripley1994] B. D. Ripley, "Neural Networks and related methods for classification (with discussion)" *J. Royal Statistical Soc. Series B*, vol. 56, pp. 409-456, 1994.

[Ripley1996] B. D. Ripley, *Pattern Recognition Ans Neural Networks*, Cambridge University Press, 1996.

[Rosenblatt1958] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the brain"; *Psychological Review*, vol. 65(6), pp. 386-408, 1958.

[Rumelhart1986] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning Internal Representations by Error Propagation", En D. E. Rumelhart y J. L. McClelland, eds., *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Bradford Books/ MIT Press, Cambridge, MA, 1986.

[Ruiz2001] A. Ruiz, P. E. López-de-Teruel, "Nonlinear Kernel-Based Statistical Pattern Analysis", *IEEE Transactions on Neural Networks*, vol. 12, pp. 16-32, 2001.

[Schapire1999] R. E. Schapire, Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions", *Machine Learning*, vol. 37, pp. 297-336, 1999.

[Schölkopf1995] B. Schölkopf, C. J. C. Burges, V. Vapnik, "Extracting Support Data for a Given Task", *Proceedings of the 1st Intlligence Conference on Knowledge Discovery and Data Mining*, (U.M. Fayyed and R. Uthurasamy, eds.), pp. 252-257; Menlo Park, CA: AAAI Press, 1995.

[Schölkopf2002] B. Schölkopf, A. I. Smola, *Learning with Kernels*, Cambridge, MA: The MIT Press, 2002.

[Shin2002] H. J. Shin, S. Cho, "Pattern Selection for Support Vector Classifiers", *Proceedings of the* 3^{rd} *International Conference on Intelligent Data Engineering and Automated Learning*, Lecture Notes in Computer Science (LNCS 2412), pp. 469-474; Manchester, UK, 2002.

[Shin2003a] H. J. Shin, S. Cho, "Fast Pattern Selection for Support Vector Classifiers", *Proceedings of the* 7^{th} *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Lecture Notes in Artificial Intelligence (LNAI 2637), pp. 376-387; Seoul, Korea, 2003.

[Shin2003b] H. J. Shin, S. Cho, "How Many Neighbors To Consider in Pattern Pre-selection for Support Vector Classifiers?", *Proceedings of the INNS-IEEE International joint Conference on Neural Networks*, pp. 565-570; Portland, OR, 2003.

[Shin2007] H. J. Shin, S. Cho, "Neighborhood Property-Based Pattern Selection for Support Vector Machines", *Neural Computation*, vol. 19(3), pp. 816-855, 2007.

[Sigillito1989] V. G. Sigillito, S. P. Wing, L. V. Hutton, K. B. Baker,"Classification of Radar Returns from the Ionosphere Using Neural Networks". Johns Hopkins APL Technical Digest, 10, pp. 262-266, 1989.

[Sklansky1980] J. Sklansky, L. Michelotti, "Locally Trained Piecewise Linear Classifiers", *IEEE Transanctions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 101-111, 1980.

[Snelson2006] E. Snelson, Z. Ghahramani, "Sparse Gaussian Processes Using Pseudo-Inputs", en *Advances Neural Information Processing Systems*, vol. 18, MIT Press, pp. 1257-1264, 2006.

[Specht1990] D. Specht, "Probabilistic Neural Networks", *Neural Networks*, vol. 3(1), pp. 109-118, 1990.

[Strand1992] E. M. Strand, W. T. Jones, "An Active Pattern Set Strategy for Enhancing Generalization while Improving Back-Propagation Training

Efficiency", *Proceedings of the International joint Conference on Neural Networks*, vol. 1, pp. 830-834; Baltimore, MD, 1992.

[Telfer1994] B.A. Telfer, H.H. Szu, "Energy Functions for Minimizing Misclassification Error with Minimum-Complexity Networks",*Neural Networks*, vol. 7, pp. 809-818, 1994.

[Van Trees1968] H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I*, Wiley, New York, 1968.

[Vapnik1995] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer Verlag, 1995.

[Vogl1988] C. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, D. L. Allcon, "Accelerating the convergence of the Back Propagation Method", *Biological Cybernetics*, vol. 59, pp. 257-263, 1988.

[Von der Malsburg1973] C. Von der Malsburg, "Self-Organization of Orientation sensitive cells in the striate cortex", *Kybernetik*, vol. 14, pp. 85-100, 1973.

[Waibel1987] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang, *Phoneme Recognition Using Time-Delay Neural Networks* , Technical Report (TR-I-0006); Japan: Advanced Telecommunications Research Institute, 1987.

[Wann1990] M. Wann, T. Hidegir, N.Ñ. Greenbaun, "The Influence of Training Sets on Generalization in Feed-Forward Neural Networks", *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 137-142; San Diego, California, 1990.

[Watson1990] G. S. Watson, "Smooth Regression Analysis", Sankhyā: *The Indian Journal of Statistics*, Series A, vol. 26, pp. 259-279, 1964.

[Werbos1974] P. J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences", Ph.D Thesis, Harvard University, Cambridge, MA, 1974.

[Widrow1959] B. Widrow, "Adaptive Sampled-Data Systems—A Statistical Theory of Adaptation", *Institute of Radio Engineers Western Electronics Show and Convention Record*, vol. 4, pp. 74-85, 1959.

[Wiener1948] N. Wiener, *Cybernetics: Or, Control and Communication in the Animal and the Machine*, Wiley, NY, 1948.

[Williams1998] C. K. I. Williams, D. Barber, "Bayesian Classification with Gaussian Processes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1342-1351, 1998.

[Xu1995] L. Xu, M. I. Jordan, G. E. Hinton, "An Alternative Model for Mixtures of Experts", *Advances in Neural Information Processing Systems*, vol. 7, MIT Press, pp. 633-640, 1995.

[Yamasaki1994] K. Yamasaki, H. Ogawa, "A Choosing Method of Training Sets which Prevent Over-Learning", in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, pp. 551-554; Orlando, FL, 1994.

[Zhang1991a] B. T. Zhang, G. Veenker, "Focused Incremental Learning for Improved Generalization with Reduced Training Sets", *Artificial Neural Networks: Proceedings ICANN*, vol. 1, eds. T. Kohonen et al, pp. 227-232, (Elsevier), 1991.

[Zhang1991b] B. T. Zhang, G. Veenker, "Neural Networks That Teach Themselves Through Genetic Discovery of Novel Examples", *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 690-685; Washington, D.C., 1991.

[Zhang1993a] B. T. Zhang, H. Mühlenbein, "Genetic Programming of Minimal Neural Nets Using Occam's Razor", *Proceedings of the International Conference Genetic Algorithms*, (Morgan Kaufmann, San Mateo), pp. 342-349, 1993.

[Zhang1993b] B. T. Zhang, "Self-development Learning: Constructing Optimal Size Neural Networks via Incremental Data Selection", Tech. Rep. No.

768, German National Research Center for Computer Science, (Sankt Augustin), 1993.

[Zhang1994a] B. T. Zhang, "Accelerated Learning by Active Example Selection", *International Journal of Neural Networks*, vol. 5, no. 1, pp. 67-75, 1994.

[Zhang1994b] B. T. Zhang, "An Incremental Learning Algorithm That Optimizes Network Size and Sample Size in One Trial", *Proceedings of the IEEE International Conference on Neural Networks*, pp. 215-220, 1994.